FAIR ALGORITHMS FOR CLUSTERING AND RECOMMENDER SYSTEMS IN UNSUPERVISED LEARNING

A Thesis Submitted

in Partial Fulfilment of the Requirements

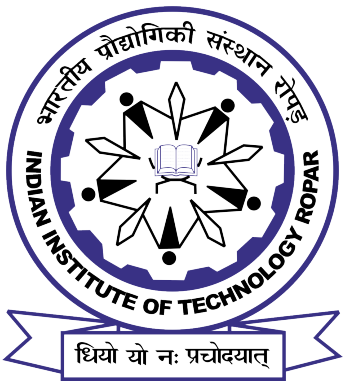
for the Degree of

DOCTOR OF PHILOSOPHY

by

Shivam Gupta

(2020CSZ0004)



DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

INDIAN INSTITUTE OF TECHNOLOGY ROPAR

April, 2025

Shivam Gupta: *Fair Algorithms for Clustering and Recommender Systems in Unsupervised Learning*

Dedicated

To my parents and sister for their unwavering love, support and belief in me.

To my supervisor for her constant encouragement and guidance throughout the journey.

Declaration of Originality

I hereby declare that the work which is being presented in the thesis entitled **Fair Algorithms for Clustering and Recommender Systems in Unsupervised Learning** has been solely authored by me. It presents the result of my own independent investigation/research conducted during the time period from September 2020 to April 2025 under the supervision of Dr. Shweta Jain, Assistant Professor, Indian Institute of Technology Ropar, Punjab, India. To the best of my knowledge, it is an original work, both in terms of research content and narrative, and has not been submitted or accepted elsewhere, in part or in full, for the award of any degree, diploma, fellowship, associateship, or similar title of any university or institution. Further, due credit has been attributed to the relevant state-of-the-art and collaborations (if any) with appropriate citations and acknowledgments, in line with established ethical norms and practices. I also declare that any idea/data/fact/source stated in my thesis has not been fabricated/ falsified/ misrepresented. All the principles of academic honesty and integrity have been followed. I fully understand that if the thesis is found to be unoriginal, fabricated, or plagiarized, the Institute reserves the right to withdraw the thesis from its archive and revoke the associated degree conferred. Additionally, the Institute also reserves the right to appraise all concerned sections of society of the matter for their information and necessary action (if any). If accepted, I hereby consent for my thesis to be available online in the Institute's Open Access repository, inter-library loan, and the title & abstract to be made available to outside organizations.

Signature

Name: Shivam Gupta
Entry Number: 2020CSZ0004
Program: Ph.D.
Department: Computer Science and Engineering
Indian Institute of Technology Ropar
Rupnagar, Punjab 140001

Date: April 2025.

Acknowledgement

Foremost, I would like to extend my gratitude towards Dr Shweta Jain, who is not just a supervisor to me but a constant motivator and inspiration for me to excel in whatever task I do. This journey would not have been possible without her guidance, support, encouragement and critical feedback on my work. Her support and experience sharing are not just limited to research; she constantly motivates us to be optimistic and have a positive attitude towards things. Her dedication to work, helping nature, and positivity inspire one to be a better individual. She is not just an advisor to me but a 'Guru' I have been blessed to have. I still remember those interview rounds and her belief in me as I was very naive to research and had just completed my bachelor's. I am fortunate that she gave me the opportunity to work with her. From the core of my heart, thank you, mam, for all those encouragements, brainstorming discussions and constant advice. I never felt like I was away from my home and parents and felt a constant maternal feeling from you. You were always a message away and available despite your other commitments. I will be indebted to you throughout my life.

The journey at IIT Ropar is incomplete without mentioning Prof. Narayanan C Krishnan, whom we call CKN sir. His charismatic aura, critical analysis and insightful feedback have been important in shaping my research work. I can not forget his classes on the mathematical foundation of CS and machine learning that helped strengthen my basics. Though I was not his student directly, he never made me feel the same; he actively made me part of his tea break discussions and gave me space in his lab. His support extended beyond the lab, and whenever we met on campus, he was always full of energy and said, 'How are you, Shivam Bhai?'. Interactions with him have always made me full of energy and motivation to work and stay happy. God also blessed me with many other mentors during my journey who have played a tremendous role in nurturing me into what I am today. I am really grateful to Dr Ganesh Ghalme for his feedback, encouraging discussions and constant inspiration to work. I am also indebted to him for providing recommendations during my initial days for the Prime Minister Research Fellowship. Special thanks for his lovely hosting during my visit to Hyderabad for the ACML conference. The journey was enlightened when I met Prof. Nandyala Hemachandra. Thank you, sir, for all the discussions. They helped me understand the importance of minor points that strengthened my work. His support extends above work, and his constant motivation and feedback boosted my morale whenever I needed them.

I also want to acknowledge the Prime Minister Research Fellowship, SERB International Travel Scheme, and ACM travel grant that enabled me to attend and present my work at different CORE A* / A conferences and prestigious events. Having mentioned conference travel, I am overwhelmed to mention Dr Shashi Shekar Jha, who has been my conference travel partner. He never made me feel alone during my stay abroad and made the trips worth remembering. He has been a constant torchbearer for advice that goes beyond my PhD and broadens my vision of life around me. I am still indebted to him for the knowledge I learned during his reinforcement learning lectures. He always had a smiling

face and warm gesture whenever we met. I would also like to thank Dr Deepti R. Bathula, Dr Arun Kumar, Dr Nitin Auluck, Dr Sujit Gujar and Dr Swapnil for their thoughtful discussions. Also, I convey my thanks and regards to Manish sir and Madhuri mam for their warm gestures.

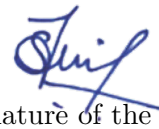
I would like to acknowledge the efforts of all my coauthors and their feedback on my work. One cannot finish this long journey without the invaluable support of family and friends. First, I would like to thank my father (Dr. O.P. Gupta) and mother, Ritu. My father has been the sole motivator in driving me throughout this journey from childhood. His constant fighting and optimistic spirit helped me sail through this journey. He was always there at my back and never let me fall. I cannot forget my mother's efforts, which shaped me into who I am today. She has been my strength and power in making me achieve my dreams. I would like to thank my sister, Deepti, for all her encouragement during my weak days. Thank you for making me cheer up and always stay happy. The thesis is incomplete without mentioning my buddies and B.Tech project partners - Jyoti Meena and Priyansh Agrawal, who have constantly pushed me to join PhD since my bachelor's. Thank you for believing in me, bearing me and being there with me till the end of my PhD journey.

Ultimately, I want to thank the Indian Institute of Technology (IIT) Ropar, Director Sir Dr. Rajeev Ahuja and Dr. Sarit Kumar Das for providing me with the environment to conduct research. I extend my thanks to the CSE office staff and academics department. Special mention to Ashu sir, Sheetal mam, Poonam mam, and Raman mam for their efficient work in the office. Special thanks to Ashu sir for his quick support whenever I needed it. I also extend my warm regards to the Heads of the Department over my PhD journey, Dr Nitin and Dr Sudarshan, for providing such an ecosystem. To conclude, a pat to myself for making it through this long journey. Cheers to those hard work, sleepless nights, honesty at work, perseverance, and commitment !!

Certificate

This is to certify that the thesis entitled **Fair Algorithms for Clustering and Recommender Systems in Unsupervised Learning**, submitted by **Shivam Gupta (2020CSZ0004)** for the award of the degree of **Doctor of Philosophy** of Indian Institute of Technology Ropar, is a record of bonafide research work carried out under my guidance and supervision. To the best of my knowledge and belief, the work presented in this thesis is original and has not been submitted, either in part or full, for the award of any other degree, diploma, fellowship, associateship or similar title of any university or institution.

In my opinion, the thesis has reached the standard of fulfilling the requirements of the regulations relating to the Degree.



Signature of the Supervisor

Dr. Shweta Jain

Department of Computer Science and Engineering

Indian Institute of Technology Ropar

Rupnagar, Punjab 140001

Date: April, 2025.

Lay Summary

The adoption of machine learning (ML) based algorithms has increased over the past few years. The main reason for the success of these machine-learning models is the growth of computational resources and digital data. The increasing data enables the algorithms to learn more generalized and complex models. Initially, the primary focus was devising ML models with improved accuracy. However, the massive adoption of highly accurate models did not go that effortlessly. There have been rising controversies about the decisions from these ML models being discriminatory to certain humans involved. These decisions can result in impartial treatment of a specific individual or group of individuals (based on attributes such as gender, income level, race or education) and have catastrophic effects. This has motivated researchers to systematically investigate and improvise the ML models before deploying them for real-world applications. The present thesis looks into developing fair machine-learning algorithms in an unsupervised setting when data has missing labels. The thesis explicitly focuses on two unsupervised algorithms: clustering and recommender systems. Firstly, clustering is a valuable machine-learning technique that identifies patterns in data by grouping similar objects (data points) based on similar characteristics. It finds applications in many fields, such as automated resume processing, employee allocation, loan approvals, etc. This thesis proposes clustering algorithms that help maintain fairness based on sensitive attributes (or protected groups) like gender, age, race, and income level. We devise clustering techniques that maintain a minimum threshold of data points from each protected group value (say male and female in gender) in every cluster. We investigate the problem in different scenarios based on the availability of data: Offline (full access), online (data arrives over time), and Federated (distributed data access). The thesis provides theoretical bounds on the performance of proposed methods and experimentally validates them on different synthetic and real-world datasets. In the other direction, the thesis explores fairness in the recommender systems by addressing biases arising in product recommendations. Typically, the recommended items fall into the popular and non-popular items category. Among these, the popular items are the ones that have been rated by many users and have existed in the systems for a long time. On the other hand, non-popular items are newer or have been rated less frequently by users. A common limitation in many of the existing recommendation models is that these models may favor popular items over time as their rating data is more evident in the dataset. This effect can amplify over time and create an unfair market where new and less-rated products face challenges for survival even though users might be interested in them. Motivated by fairness in clustering methods where we balance data points from different group values, the thesis looks into ensuring a fair opportunity for both popular and non-popular items. We propose a fair recommendation algorithm that mitigates popularity bias and empirically validates its efficacy on various real-world datasets against existing state-of-the-art methods.

Abstract

With the advent of technology, the adoption of Artificial Intelligence (AI) and Machine Learning (ML) based decision systems into daily human life has significantly increased. Recent studies have exposed the prejudiced outlook (biasness) in the ML outcomes towards individuals and groups of individuals characterized through protected attributes such as race and gender. These decisions have a direct and long-lasting impact on the humans involved. Fairness has gained considerable attention from the research community when data labels are available for prediction modelling, i.e., supervised learning. However, in real-life scenarios, data may lack labels and providing manual labels will require proper incentivization or expertise. Consequently, researchers have started exploring fairness issues in unsupervised learning, which forms the focus of this thesis. In particular, the primary focus of this thesis is to address both theoretical underpinnings and practical implications of fair algorithms for unsupervised learning in the context of clustering and recommender systems. The contributions of the thesis include:

1. **Group Fair Notions and Algorithms in Offline Clustering**: The thesis first theoretically establishes relationships between different existing discrete group fairness notions and then proposes a generalized notion of group fairness for multivalued group values. We propose two simple and efficient round-robin-based algorithms for satisfying group fairness guarantees. We next prove that the proposed algorithm achieves a two-approximate solution to optimal clustering and show that the bounds are tight. The efficacy of the proposed algorithms is also shown via extensive simulations.

2. **Nash Social Welfare for Facility Location**: To investigate the problem of satisfying multiple fairness levels simultaneously, the thesis extends the fair clustering problem to the facility location problem. The thesis proposes the first-of-its-kind application of modelling Nash Social Welfare for facility location to target multiple fairness while focusing on minimizing the distance between individuals. The proposed polynomial time algorithm works for any h-dimensional metric space and allows facilities to be opened at a specified set of locations rather than solely at the individuals' own locations, as in most previous literature. The proposed algorithm provides a solution that satisfies group fairness constraints and achieves a good approximation for individual fairness. The proposed method undergoes real-world testing on the United States (US). census dataset, with road maps providing the actual car road distances between individuals and facilities.

3. **Group Fairness in Online Clustering**: To tackle the challenge of handling group fairness requirements in an online model, the thesis proposes a randomized algorithm that prevents the over-representation of any protected group by applying capacity constraints on the number of data points from each group that can be assigned to a particular cluster. The proposed method achieves a constant-cost approximation to

optimal offline clustering and also handles the challenge of an apriori unknown total number of data points using a doubling trick. Empirical results demonstrate the proposed algorithms' efficacy against baseline methods on synthetic and real-world datasets.

4. **Fairness in Federated Data Clustering**: For addressing fairness in distributed settings, the thesis analyzes federated data clustering to ensure privacy-preserving clustering in a distributed environment. The proposed method results in cluster centers with lower cost deviation across clients, leading to a fairer and more personalized solution. The method is validated on different synthetic and real-world datasets, with results demonstrating effective performance against state-of-the-art methods.

5. **Popularity Bias in Recommender System**: While the first four contributions focus more on clustering. This contribution primarily analyzes the fairness aspects of recommender systems. The thesis proposes a novel metric that measures popularity bias as the difference in the Mean Squared Error (MSE) on the popular and non-popular items. Further, we propose a novel technique that solves the optimization problem of reducing overall loss with a penalty on popularity bias. It does not require any heavy pre-training and undergoes extensive experiments on real-world datasets displaying outperforming performance on recommendation accuracy, quality, and fairness.

Keywords: Fairness; Unsupervised Learning; Clustering; Group Fairness; Online Algorithms; Federated Learning; Recommender Systems; Matrix Factorization; Popularity Bias;

List of Publications

Journals

[**J1**]. **Shivam Gupta**, Ganesh Ghalme, Narayanan C. Krishnan, and Shweta Jain. Efficient Algorithms for Fair Clustering with a New Notion of Fairness. Data Mining and Knowledge Discovery, pages 1–39, 2023 **(Impact factor 5.406)** (**Best Paper Award** 🏆 at International Conference on Deployable AI 2022).

[**J2**]. **Shivam Gupta**, Kirandeep Kaur, and Shweta Jain. EqBal-RS: Improved Matrix Factorization for Mitigating Popularity Bias in Recommender Systems. Journal of Intelligent Information Systems, pages 1–26, 2023. **(Impact Factor 3.4)**

Conference Proceeding

[**C1**]. **Shivam Gupta**, Ganesh Ghalme, Narayanan C Krishnan, and Shweta Jain. Group Fair Clustering Revisited – Notions and Efficient Algorithm. In International Conference on Autonomous Agents and Multiagent Systems (AAMAS), pages 2854–2856, 2023. **(CORE A*)**.

[**C2**]. Jaglike Makkar, Bhumika, Shweta Jain, and **Shivam Gupta**. `MFC`: A Multishot Approach to Federated Data Clustering. European Conference on Artificial Intelligence (ECAI), pages 1672 – 1679, 2023. **(CORE A)**.

[**C3**]. **Shivam Gupta**, Shweta Jain, Narayanan C. Krishnan, Ganesh Ghalme, and Nandyala Hemachandra. Online Algorithm for Clustering with Capacity Constraints. In Joint International Conference on Data Science & Management of Data (CODS-COMAD), 2024. (Extended Abstract) (**Best Paper Award** 🏆 Runner's up in young researcher's symposium at the conference).

[**C4**]. **Shivam Gupta**, Shweta Jain, Narayanan Krishnan, Ganesh Ghalme, and Nandyala Hemachandra. Capacitated Online Clustering Algorithm. European Conference on Artificial Intelligence, 2024. **(CORE A)**.

Book Chapter

[**B1**]. **Shivam Gupta**, Shweta Jain, Ganesh Ghalme, Narayanan C Krishnan, and Nandyala Hemachandra. Group and Individual Fairness in Clustering Algorithms. In Ethics in Artificial Intelligence: Bias, Fairness and Beyond, pages 31–51. Springer, 2023.

Workshops

[**W1**]. **Shivam Gupta**, Ganesh Ghalme, Narayanan C Krishnan, and Shweta Jain. Group Fair Clustering Revisited – Notions and Efficient algorithm. Workshop on Games, Agents and Incentives (GAIW), AAMAS, 2023. **(CORE A*)**.

Under Review Articles

[**J3**]. **Shivam Gupta**, Tsering Wangzes, Tarushi, and Shweta Jain. Fair Federated Data Clustering through Personalization: Bridging the Gap between Diverse Data Distributions. arXiv:2407.04302, 2024.

[**C5**]. **Shivam Gupta**, Avyukta Manjunatha Vummintala, Shweta Jain, and Sujit Gujjar. Balancing Fairness and Efficiency in Facility Location via Nash Welfare Perspective.

Other publications during Ph.D. (Not a part of the thesis)

[**J4**]. Ranjana Roy Chowdhury, **Shivam Gupta**, and Sravanthi Chede. World War III Analysis using Signed Social Networks. Social Network Analysis and Mining, 11:1–16, 2021. **(Impact factor 3.87)** (Equal Contribution).

[**C6**]. Manan Singh, Sai Srinivas Kancheti, **Shivam Gupta**, Ganesh Ghalme, Shweta Jain, and Narayanan C. Krishnan. Algorithmic Recourse based on User's Feature-order Preference. In Joint International Conference on Data Science & Management of Data (CODS-COMAD), pages 293–294, 2023. (Extended Abstract).

[**C7**]. Rejoy Chakraborty, Chayan Halder, Kaushik Roy, **Shivam Gupta**, and Shashi Shekar Jha. MSBNet: Handwritten Bangla Character Recognition using Lightweight Multi-scale CNN Architecture. In Joint International Conference on Data Science & Management of Data (CODS-COMAD), 2024.

[**C8**]. Avyukta Manjunatha Vummintala, **Shivam Gupta**, Shweta Jain, and Sujit Gujjar. FLIGHT: A Unified Framework for Symmetric and Asymmetric Welfares for Facility Location. International Conference on Autonomous Agents and Multiagent Systems, 2025. (Accepted) **(CORE A*)**.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

"Whatever affects one directly, affects all indirectly" — Martin Luther King Jr.[1]

"The test of our progress is not whether we add more to the abundance of those who have much; it is whether we provide enough for those who have too little."
— Franklin D. Roosevelt (Former President of the United States)[2].

"Artificial Intelligence is a tool. The choice about how it gets deployed is ours."
— Oren Etzioni (Former CEO, Allen Institute for Artificial Intelligence)[3].

1.1 Introduction

With the advent of technology, the adoption of Artificial Intelligence (AI) and Machine Learning (ML) based decision systems into daily human life has significantly increased. Today, we are surrounded by ML-based models in a variety of applications, including loan and insurance approvals, college admissions, job hiring, government recidivism, etc. Traditionally, the primary goal of ML algorithms in these systems has been to achieve the highest possible accuracy and predictive performance. However, examining their outcomes more closely becomes crucial when ML algorithms are applied in areas that impact society. Recent studies have exposed the prejudiced outlook (biasness) in the ML outcomes [15, 16] towards individuals and groups of individuals characterized through protected attributes such as race and gender. These decisions have a direct and long-lasting impact on the humans involved.

Figure 1.1 illustrates several anecdotes observed in recent years about the social effects of ML applications. The most widely discussed controversy among these in ML literature is the COMPAS controversy. COMPAS (Comprehensive Online Management Personnel and Accounting System) is a criminal risk prediction (or scoring) system introduced in United States (US) court trials. A report by *ProPublica* revealed that the system was biased towards African Americans (Non-white) [7]. For example, a non-white American committed a crime in her juvenile, and after bail, the person did not commit any criminal offences. However, the system marked him/her as more risky compared to a white who re-committed serious crimes after serving his/her punishment and being released on bail. Similarly, credit cards powered by Apple and Goldman Sachs were biased towards the

[1] https://www.africa.upenn.edu/Articles_Gen/Letter_Birmingham.html
[2] https://www.loc.gov/item/today-in-history/january-20/
[3] https://www.techtarget.com/searchnetworking/feature/Whats-the-status-of-AI-in-networking

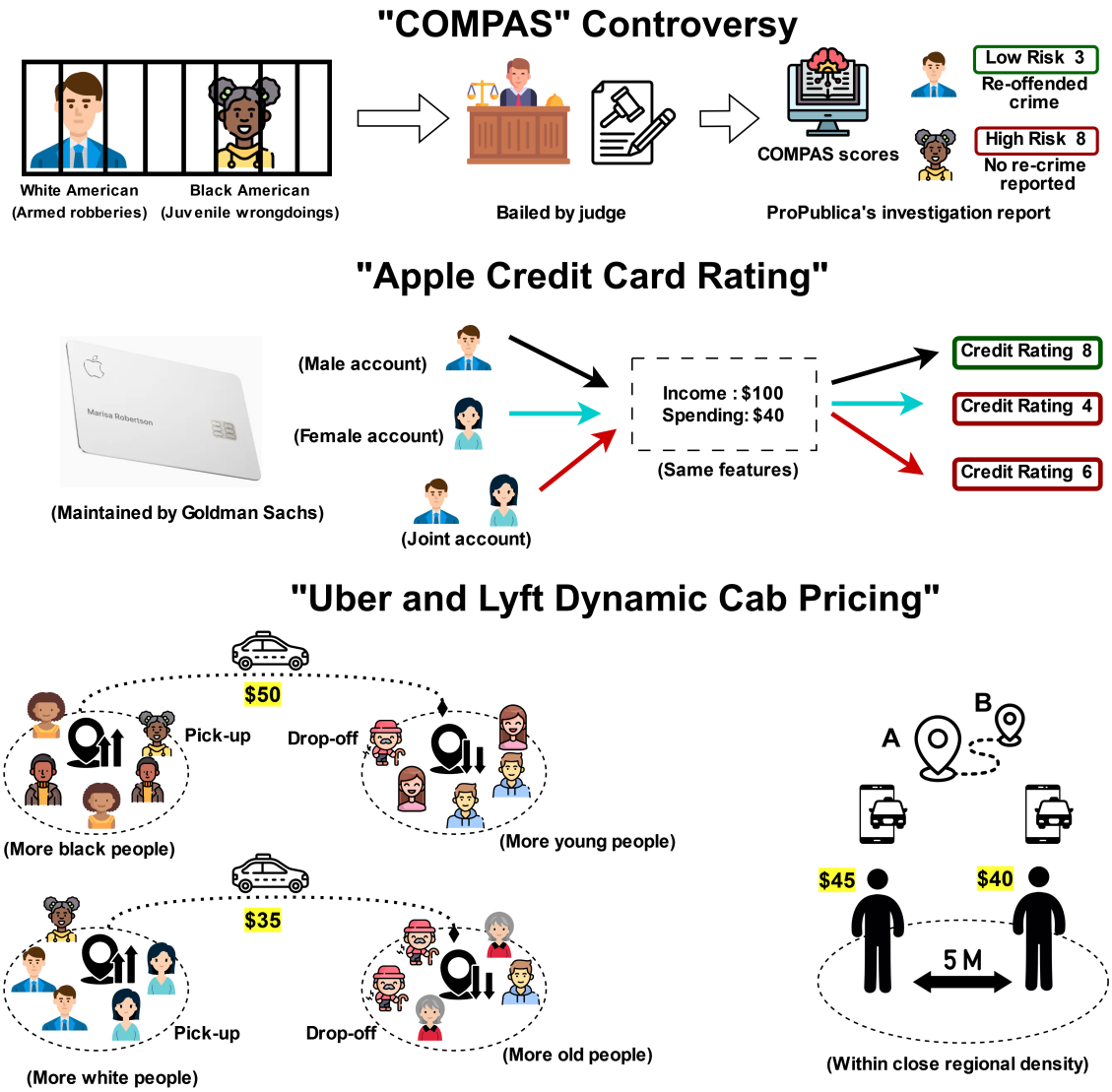
presence of females in their bank account holdings. Individual female account holders (or joint account holders with females) were assigned lower credit scores than solely male individuals (or joint accounts within male members), even when all shared the same features regarding savings and expenditure [3]. Also, recently, it became evident in dynamic cab pricing platforms such as Uber and Lyft that these services charged higher bucks if the customer's drop-off or pick-up location was in a region dominated by a non-white community [5]. Note that bias is not limited to racial boundaries but includes features such as education level and age [6] etc. Additionally, studies have reported biased decision-making in the termination of drivers registered with Uber and Lyft, with a disproportionate number of those dismissed belonging to marginalized groups.

In all the above instances, unfair behaviour revolved around the presence of certain sensitive (or protected) groups such as gender, race, etc. However, controversies are observed at individual levels as well. For instance, two customers within a close regional density were offered different pricing for the same drop-offs even when they confirmed their booking almost simultaneously [4]. Other evidence-reporting unjust decisions include face recognition [2] and targeted advertisement systems [1]. These were prejudiced towards certain demographic groups or brands respectively. Thus, designing fair and accurate ML models is central to improving the models' trustworthiness [17].

Fairness has gained considerable attention from the research community when data labels are available for prediction modelling, i.e., supervised learning [18, 19, 20]. However, in real-life scenarios, data may lack labels and providing manual labels will require proper incentivization or expertise [21]. Consequently, this thesis focuses on exploring fairness issues in unsupervised learning. In particular, we look into clustering and recommender systems. Among these, clustering deals with dividing the data points into groups (called clusters). The clusters are so formed that data points within the same cluster are more similar than the others. Alternatively, recommender systems are ML models that suggest individuals (or users) with a set of items (or products) they might prefer based on their history or preferences. We now present four motivating examples to help understand the importance of fairness in unsupervised learning, particularly clustering and recommender systems.

1.1.1 Motivating Examples: Need for Fairness in Unsupervised Learning

- **Employee Allocation**- Consider an employee-friendly company looking to open branches at multiple locations across the country and distribute its employees in these branches. The company has employees with diverse backgrounds based on race, gender, etc., and does not prefer any group of employees over other groups based on these attributes. Where should a company open branches, and how should employees be allocated to minimize travel distance? This problem can be naturally solved as a clustering problem (Figure 1.2). However, several more open questions need investigation: Where should a company set up branches that maintain diversity



"Face Recognition"

Classifier	Overall Accuracy	Darker Males	Darker Females	Lighter Males	Lighter Females
Microsoft	93.7%	94.0%	79.2%	100%	98.3%
Face++	90.0%	99.3%	65.5%	99.2%	94.0%
IBM	87.9%	88.0%	65.3%	99.7%	92.9%

Figure 1.1: Figure shows different controversies that are reported in real-world ML deployments [1, 2, 3, 4, 5, 6, 7].

in terms of a minimum fraction of employees from each group (say race)? What changes will be in clustering assignments or additional distance that employees must travel to help the organization achieve these diversity constraints? Will fairness increase the existing hardness of the clustering problem? How will the company handle the continuous influx of new employees without significantly impacting the assignments of previous employees? Can organizations analyze data spread (or distributed) across different sites (or databases) while preserving privacy yet still resulting in highly accurate and reliable clusters? Will maintaining diversity hamper individual expectations?

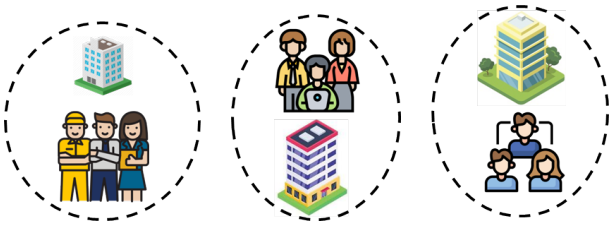


Figure 1.2: Figure shows the use of fair clustering for achieving male-to-female diversity ratio of say 2 : 1 in every office.

✳ **Wholesale Distribution Network** Consider the dynamic landscape of wholesale distribution networks. In such a scenario, retailers employ salespersons who navigate cities to promote products, offer discounts, and build relationships with individuals (or consumers) [22]. To enhance consumer retention in wholesale distribution networks, it becomes imperative to provide specialized salespersons for marketing. Clustering offers a promising solution to achieve such an efficient market coverage by forming clusters of consumers (shopkeepers and direct customers) based on various features such as product consumption, order volume, and location [22]. The resulting clusters group similar consumers together for personalized marketing (see Figure 1.3). However, as each cluster provides certain marketing discounts and offers, it becomes crucial to avoid customers feeling biased. Thus, no cluster should be over-represented by customers from a particular group value (say based on income) while handling the continuous influx of customers. Also, one should note that this needs to be achieved while maintaining a healthy work-life balance for salespersons to maintain quality service.

✳ **Bank Loan Approvals** Consider a scenario where a bank needs to analyze multiple loan applications based on features such as income, loan purpose, credit score, age, number of dependents, etc. [23, 24, 25]. In such cases, clustering can be a useful method to assist bank managers in analyzing thousands of applications. By clustering similar individuals (data points), managers can thoroughly check a few representative applications (cluster centers) from each cluster and apply the same decision to all cluster members (see Figure 1.4). Now, the set of questions that arise are as follows - Can banks maintain a minimum fraction of each group in every cluster

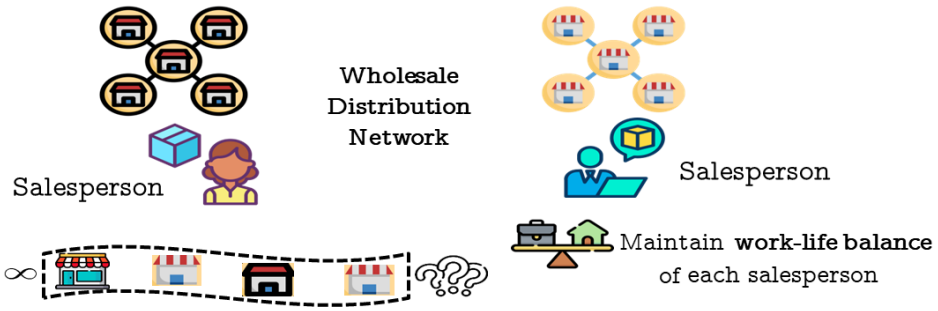


Figure 1.3: Fair clustering in the wholesale distribution network for personalized marketing.

to ensure a fair approval process? How can banks adapt to handle the continuous influx of applications? Can multiple banks come up to develop a single model to differentiate legitimate applications from fraudulent ones [26]?

Figure 1.4: Example illustrating application of clustering for speeding up loan approval process in banks.

- **Recommendation Systems** on online platforms, such as movie streaming services, search engine advertisements, and e-commerce websites have reported facing fairness issues [8, 27, 28]. For example, in product recommendation, a set of few items are frequently rated by users (aka. popular items), while a long tail distribution of items (called non-popular items) are either new or less-rated by users. Now, consider the recommender system model that recommends popular items to most users, even if they are less preferred, while new or non-popular items starve for desired visibility. Such a practice can create exclusive market positions for certain items, posing challenges for firms and stifling innovation in product development. For instance, recently, Google's targeted advertisements (Google Ads) were found to favor popular brands [1, 29].

To summarize, the primary focus of this thesis will be to provide a formal and comprehensive answer to these questions, addressing both theoretical underpinnings and practical implications of fair algorithms for unsupervised learning in the context of clustering and recommender systems. We now briefly discuss both techniques to better understand and position the thesis in upcoming sections.

1.2 Clustering

Clustering deals with partitioning a set of data points into groups (called clusters), with each cluster being represented by a cluster center. The goal of any centroid-based clustering algorithm is to minimize intra-cluster similarity (or maximize inter-cluster[4] dissimilarity) between data points and center[5]. Since labels are absent for similarity comparison in unsupervised learning, the choice of similarity metric becomes crucial. The most commonly used similarity metric are distance metrics such as Euclidean (2-norm), Manhattan (1-norm) distance, etc [30]. The choice of different distance metrics results in calling clustering methods as k-means for 2-norm, k-median for 1-norm and k-center for infinity norm[6]. Also, the sum of distances between data points and corresponding cluster centers is called *clustering objective cost* [31].

Note that in traditional clustering methods, a data point belongs to a single cluster deterministically, i.e., non-fuzzy (hard assignments) and is the focus of this thesis. Since clustering involves assigning data points to a different cluster, the number of such possibilities increases exponentially as the number of data points and centers increases. Thus, the clustering problem is proved to be NP-hard [32, 33, 34, 35]. Despite NP-hardness, many heuristics and approximation algorithms exist and are widely used in real-world applications [30]. The best-known approximation factors for k-means [32], k-median [33] and k-center [35] are $2, (1 + \sqrt{3} + \epsilon)$ and 2 respectively for small constant $\epsilon > 0$. Next, we now categorize clustering algorithms based on accessibility to data points and applications.

1.2.1 Categorization based on Data Accessibility and Application

1. **Offline Clustering**– In offline clustering, all the data points are known in advance and are available in memory. This model provides the most flexibility in terms of data availability. However, the scalability of these offline solutions is constrained by the size of the main memory [30, 31].

2. **Streaming Clustering**– In contrast, when the number of data points exceeds the size of the main memory, streaming environments divide the data into chunks. The size of chunks is chosen so that they can easily fit into the main memory. A complete iteration over all the chunks is said to be one read (or pass). Each read involves processing the chunk and storing small information out of it for later use. The clustering results are obtained at the end of one or more full reads. Thus, the efficacy of these methods depends on the number of passes that need to be performed over the complete data [36, 37].

3. **Online Clustering**– A more stringent variation of offline and streaming setting is online clustering, where an endless stream of data points arrives over time.

[4] intra-cluster refers to within cluster and inter-cluster refers to between different clusters.
[5] The terms cluster centers and centers are often used interchangeably for simplicity and ease of reading.
[6] It is measured as the maximum absolute difference between corresponding components of two vectors.

Due to limited memory, the algorithm must make an irrevocable decision about incorporating an incoming data point into existing clusters or opening it as a new center. Once a data point becomes a center, it remains so forever. Similarly, any data point previously seen cannot be chosen as the center when a new data point arrives. An important aspect to note in online clustering pertains to the absence of information regarding the ordering of the arrival of points in the stream. As a result, the algorithm ends up opening more number of centers than the desired target to maintain good approximation guarantees on objective cost [38, 39].

4. **Federated Data Clustering**– In this setting, the data points are spread across different clients. The goal is to find a set of global centers that best partition each client's local data points. As federated is a privacy-preserving distributed setup, therefore it is forbidden to share the original data points between the client and server. The algorithms can only share limited information, such as best centers on local data or synthetic data [40, 41]. It is important to note that the primary challenge in this setting is that clients may not contain data from all the true unknown (say, k) clusters.

1.2.2 Fairness Levels in Clustering

The fairness in clustering is under investigation from different perspectives depending upon the real-world application requirements. We refer to these perspectives as different levels of fairness and now discuss some of the widely studied levels in fair clustering literature.

1. **Group Fairness**– The prevalence of anthropological factors such as discrimination based on gender, race, and ethnicity in the data has resulted in a plethora of techniques to achieve group fairness. Group fairness demands that different groups should be treated in an unbiased manner. For instance, discrimination, such as shortlisting fewer qualified females for high-paying jobs, is unfair to the female group. In the clustering context, group fairness techniques focus on achieving approximately equal (or user-desired) representation for all protected group values (say male and female for gender) in every cluster [42, 43, 44, 45, 46]. To mathematically model group fairness into clustering, different group fairness notions (or metrics) have come up in the past literature. The main purpose of these notions is to capture the user's desired requirements of fairness level.

2. **Individual Fairness**– Group fairness does not ensure fair treatment for a particular individual. The trait of human envy might still make an individual discontented. For example, an employee might feel discriminated against or left out if similar employees receive a favorable appraisal. There are algorithms in the literature that guarantee individual fairness by focusing on the principle that similar individuals in the context of a particular task should receive similar outcomes [15].

3. **Social Fairness**– deals with the biasness arising when outcomes from an algorithm

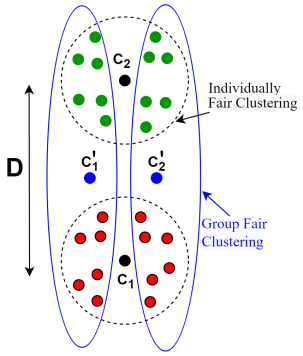


Figure 1.5: Figure shows conflicting nature of group and individual fairness. Here C_1, C_2 are individual fair centers separated by large distance D, and C'_1, C'_2 are group fair centers.

can be highly unfavourable for some protected (or sensitive) values. For example, say, employees belonging to the racial group value (say Asian-Pacific) have to travel an average more distance compared to other racial group values. It is important to note that group fairness ensures a minimum fraction of assignments in each cluster from every protected group value. That is, fairness constraints are applied at the cluster level by imposing constraints on the proportions of groups in each cluster. On the contrary, social fairness tries to provide a more equitable overall average cost for different protected group values irrespective of cluster assignments [47, 48].

4. **Diverse-Center Selection Fairness**– In many real-world applications, such as data or news summarization, the input for downstream tasks consists of data points chosen as cluster centers. For example, instead of displaying all matching images in an image database, it is beneficial to show only the summary (centers) obtained by clustering for a given query. Recent evidence shows that such summarization can sometimes be quite unfair to certain protected group values. For instance, it is observed that Google Image search for CEOs resulted in a higher proportion of male than female images (cluster centers). However, in the real world, females comprise around 30% of CEOs worldwide [49, 50]. Therefore, maintaining diversity at the center level becomes crucial in such applications. To this, diverse center selection ensures that the clustering output maintains a lower and upper bound on the number of centers to be chosen from every protected group value [51, 52, 53].

5. **Additional Fairness Levels** – Recent efforts have proposed fairness from alternative perspectives, but the available literature is still in its infancy. These include proportional [54, 55, 56, 57, 57, 54] or core fairness [58], which allows a subset of data points bearing minimum size constraint to choose a better center (if it exists) to lower the cost. Similarly, representative fairness ensures centers are closer to data points [59, 60, 61, 62], and pairwise fairness ensures the probability of a pair of data points belonging to different clusters varies based on the distance between them [63].

1.2.3 Balancing Multiple Fairness Levels

Many real-world applications demand the need to satisfy multiple levels of fairness simultaneously. However, it is not always necessary that different levels of fairness will go hand in hand. They might become contradictory to each other, i.e., satisfying one level of fairness might degrade the efficacy of the other. Recent attempts have been made in this direction [64, 65] and observations are discussed below:

- **Group and Individual Fairness**– The two fairness levels arose independently in fair clustering literature. However, in real-world applications such as direct marketing, the corporate house's diversity policy necessitates group fairness. Simultaneously, customers might feel discontented if people in their similarity set belong to a different cluster than their own (hence offering different benefits). Thus, there is a need to study the relationship between the two levels. Recent attempts [64, 65] explore this direction and propose instances that show the conflicting nature of both the fairness levels, i.e., satisfying one might adversely affect the other. To understand this, consider a dataset with data points split across two far-apart clusters, each containing points from one protected group (as illustrated in Figure 1.5). Group fair clustering will try to place the cluster centers in between the two clusters. On the contrary, the original cluster centers will also serve as optimal individual fair centers when the individual fairness notions depend on density-based similarity. Thus, showing both levels of fairness as conflicting problems. The thesis explores this direction and provides evidence that both fairness levels are not strictly conflicting on real-world datasets. Further, algorithms exist (proposed contribution) that satisfy strict group fairness while still inducing a certain level of individual fairness in the clusters.

- **Group and Diverse-Center Selection Fairness**– Recent attempt by Dickerson et al. [51] theoretically shows that imposing group fairness on a solution satisfying diverse-center selection can result in a bounded increase in clustering cost. It is also subject to the condition that an additive violation of two is allowed in group fairness constraints. On the other hand, the reverse may not be true, as enforcing diverse-center selection on a solution obeying group fairness may lead to an unbounded increase in objective cost for instances with more than two cluster centers.

- **Individual (or Social) Fairness with Group Fairness (or Diverse-Center Selection)**– Dickerson et al. [51] also shows that there exist instances under certain conditions such that individual (or social) fairness are incompatible with group fairness. Similarly, the authors show that each individual and social fairness are incompatible with diverse-center selection.

1.3 Recommender Systems

With the advent of technology, online services such as movie and music streaming platforms or e-commerce websites have increased. These services offer individuals with an abundance of options (or products) to choose from. Also, individuals now have the possibility to add their feedback and ratings about the products (or services) they buy (or use). Therefore, this creates a *choice overload problem* for the individuals who want to select the best products that are suited to their needs. To address this challenge, Recommender Systems has offered a promising solution over the past decade. Primarily, recommender systems are machine learning models that suggest individuals (users) with a set of items (products) based on their past history and behaviour. The performance of any recommender systems model is evaluated on the basis of its accuracy regarding the products it recommends to users [66, 67]. One of the most crucial aspects to improve the accuracy of recommender systems is feedback, which is captured broadly in two forms: implicit feedback and explicit feedback. The implicit feedback includes click-stream data, purchase history, or time spent on an item. However, implicit feedback may not always clearly indicate user preferences. Users may click on items for various reasons, such as curiosity or price comparison, without being interested. On the other hand, explicit feedback is a more reliable [68] and accurate estimate of the user's interest [69]. It captures the absolute preferences in the form of a rating on a defined scale, such as one to five[7]. The past literature in recommender systems has investigated both feedback ratings [70, 71, 8, 27]. However, recent studies have reported that the training data capturing user-item feedback, if not captured properly, can become a potential source of biases in the recommender systems model. The main reason behind this is that the training data often consists of an uneven distribution for users and items. In other words, there exists only a small subset of users (known as popular users) who provide feedback for most of the items (see Figure 1.6, Right Side). Similarly, there exists a subset of items (called popular items) that receive the majority of ratings, and the long tail distribution of items (called non-popular items) are either new or less rated by users (see Figure 1.6 (Left Side)). The presence of such a long-tail distribution can result in ***popularity bias*** as the recommender systems model can learn to focus on these highly rated items and users. We will now discuss the aspects of popularity bias from both the item and user sides.

1.3.1 Fairness in Recommender Systems

Recommender systems suffer from many biases, such as conformity bias, inductive bias, etc. [72], but for this thesis, we will focus on popularity bias. We now look into the different categorizations of fairness in recommender systems in the context of popularity bias.

[7]The scale of one to five (or ten) is most common in many real-world use cases, but designers are free to choose any scale. Further, it depends upon the implementation whether the scale value of one is the lowest or highest rating. But the common notion considers one as the lower rating.

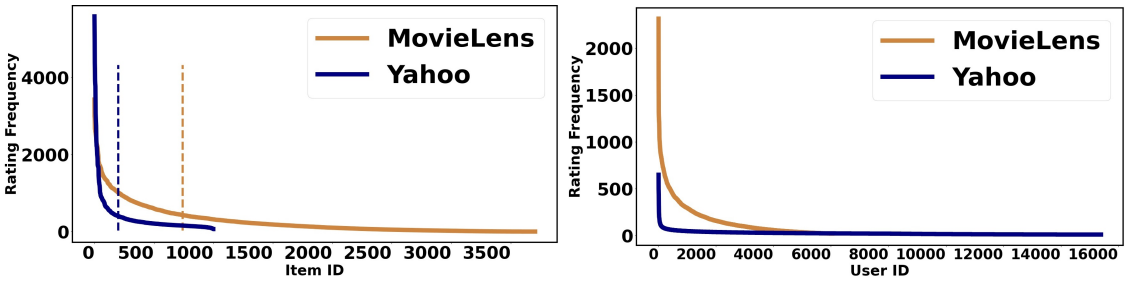


Figure 1.6: (Left): Long tail distribution in ratings for different items arranged in descending order of frequency on famous MovieLens and Yahoo datasets. (Right): Long tail distribution in the number of users who rate different items on famous MovieLens and Yahoo datasets [8]. (Best viewed in color).

1.3.2 Categorization of Fairness in Recommender Systems

In order to understand the fairness aspects in the paradigm of popularity bias, let us consider an investigating example on the real-world MovieLens and Yahoo dataset. MovieLens is a movie rating dataset with 1 million (100K) ratings given to 3706 movies by 6040 users, and Yahoo is a music rating data repository with $365,000$ ratings for 1000 songs rated by $15,400$ users.

1. **Item-side Fairness**– Recent studies have observed an imbalance in the efficacy of recommender systems on different items. For instance, consider the following toy experiment – If we group items based on the number of ratings they have received from different users and consider top 20% items as popular items and all others as non-popular items. Then, it is evident from the results reported in Figure 1.7 on famous recommender systems algorithms such as Matrix Factorization (MF) that mean square training loss is more for non-popular items than popular items. The reason is that popular items appear more frequently in training data and receive more exposure in objective functions. Such a bias is known as popularity bias on the item side and is the focus of the current thesis [8, 27].

2. **User-side Fairness** – On parallel lines, studies report an imbalance in performance for popular and non-popular users. This can also be validated by our study of the mean square loss of popular users versus non-popular users on the MovieLens[8] and Yahoo [9] dataset (Figure 1.7). We consider the first 20% users in the rating frequency plot (Figure 1.6) as popular users and execute results for the MF algorithm. The line of research mitigating popularity bias from this perspective falls into user-side fairness literature [28, 73, 74, 75].

3. **Item and User-side Fairness** – A few recent works attempt to simultaneously handle popularity bias from both item-side and user-side perspectives. One can look into the works by Liu et al. [76], Elahi et al. [77] and references therein for details.

[8]https://grouplens.org/datasets/movielens
[9]https://webscope.sandbox.yahoo.com

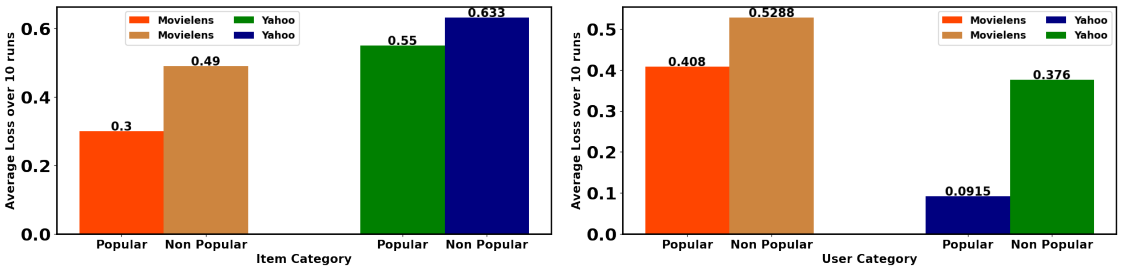


Figure 1.7: Plots show loss values in the Matrix Factorization method [9] for explicit recommendation when trained on Movielens and Yahoo dataset for (Left): Different Item types, namely popular and non-popular. (Right): Different types of users, namely popular users and non-popular users. (Best viewed in color).

1.4 Research Objectives

Building on the previous discussion, we now outline the research objectives (***Obj.***) that will form the focus of this thesis.

[Obj 1.] (a) Study the relationship between existing group fairness notions. Develop a generalized group fairness notion for multi-valued protected groups (say race) and handle the user-desired level of group fairness. (b) Devising polynomial time algorithm for k-means, k-median, and k-center clustering. (c) Theoretically bound the objective cost approximation factor of proposed algorithms.

[Obj 2.] Study the practical implications of handling multiple levels of fairness. Devise an algorithm for simultaneously achieving good approximation to both group and individual fairness.

[Obj 3.] Develop an algorithm for online clustering under group fairness constraints. Provide the bounds on the cost approximation factor and the number of centers that need to be opened.

[Obj 4.] Addressing fairness issues in privacy-preserving distributed fair clustering setting, i.e., federated clustering. Developing a global clustering strategy that is fair across all clients irrespective of distance metric (1-norm, 2-norm or infinity norm). Analyze the effect of the division of data points across clients on the algorithm's performance.

[Obj 5.] Investigate fairness aspects, similar to group fairness relevant in recommender systems. Proposing a fair algorithm that outperforms existing state-of-the-art (`SOTA`) performance on real-world datasets.

1.5 Positioning and Contribution of Thesis

The thesis deals with fair algorithms for unsupervised clustering and recommender systems. Figure 1.8 and 1.9 categorize fairness on different levels as discussed in Section 1.2.2 and 1.3.2 for clustering and recommender systems, respectively. It further brings out

the positioning of our work with respect to the existing literature. The figure indicates that there are many open areas for future exploration as well. In particular, the following are the contributions (**Contri.**) of this thesis.

[**Contri 1.**] **(Chapter 3)** The thesis establishes the theoretical relationship between different existing group and individual fairness notions. Further, a generalized notion of group fairness for multi-valued group values called $\boldsymbol{\tau}$-ratio fairness is proposed. The relationship undergoes empirical validation as well as benchmarking real-world datasets. Further, two simple and efficient round-robin-based algorithms for satisfying $\boldsymbol{\tau}$-ratio fairness are proposed, namely FRAC and FRAC$_{OE}$. The algorithms allow a user-specific level of group fairness and incur only an additional time complexity of $O(kn \log n)$, best in current literature. Here, n is the total number of data points that need to be partitioned into k clusters. The experimental efficacy of both methods is validated on four real-world datasets for k-means and k-median. Also, the thesis reports theoretical guarantees for FRAC$_{OE}$.

[**Contri 2.**] **(Chapter 4)** The thesis studies the problem of simultaneously satisfying multiple levels of fairness. The thesis proposes the first-of-its-kind application of modelling Nash social welfare instead of considering standard utilitarian or egalitarian approaches to target multiple fairness. We propose an efficient and scalable algorithm called `FAIRLOC` that minimizes the product of distances of data points to assigned centers while obeying group fairness constraints. We theoretically provide approximation bounds on cost with respect to optimal fair allocation and show that `FAIRLOC` achieves a quadratic approximation in the product-based objective function. The thesis conducts near real-world testing of `FAIRLOC` on United States census datasets. The results showcase that `FAIRLOC` provides a solution with significantly lower costs and better group and individual fairness metrics than state-of-the-art methods.

[**Contri 3.**] **(Chapter 5)** To tackle the challenge of handling group fairness requirements in an online model, the thesis proposes a randomized algorithm that prevents the over-representation of any protected group. This is ensured by applying capacity constraints on the number of data points from each group that can be assigned to a particular cluster. The proposed methods achieve a constant-cost approximation to optimal offline clustering and handle the challenge of an apriori unknown total number of data points using a doubling trick. Empirical results demonstrate our method's efficacy against `SOTA` methods on various synthetic and real-world datasets.

[**Contri 4.**] **(Chapter 6)** For addressing fairness in distributed settings, this thesis analyzes federated data clustering to ensure privacy-preserving clustering in a distributed environment. We first propose a federated data clustering method called `MFC`. The method achieves data distribution independence and has a theoretical bound on the quality of centers obtained. We further extend it to propose `p-FClus`results in

cluster centers with lower cost deviation across clients, leading to a fairer and more personalized solution. The method is the first attempt to provide personalization in federated data clustering. Furthermore, `p-FClus` achieves a lower clustering objective cost in a single communication round between the server and clients, regardless of the nature of data distribution (or division) among clients. The method is validated on different synthetic and real-world datasets, with results demonstrating effective performance against `SOTA` methods.

[**Contri 5.**] **(Chapter 7)** While the first four contributions focus more on clustering. This contribution primarily analyzes the fairness aspects of recommender systems. The thesis proposes a novel metric, Popularity Parity, that measures popularity bias as the difference in the Mean Squared Error (MSE) on the popular and non-popular items. Further, EqBal-RSis proposed, a novel technique that solves the optimization problem of reducing overall loss with a penalty on popularity bias. It does not require any heavy pre-training and undergoes extensive experiments on real-world datasets displaying outperforming performance on recommendation accuracy, quality, and fairness. The method works exceptionally well on Popularity Parity while having comparable performance on prior existing metrics and does not compromise on the diversity of items.

1.6 Organization of Thesis

The thesis consists of eight chapters, each addressing different aspects of fairness in unsupervised learning. Chapter 1 serves as an introduction, where we present the problem of fairness in unsupervised clustering and recommender systems. We explore different real-world examples that necessitate the need for fair algorithms. We then formally define different existing formulations for handling biases and explore different setups considered in the thesis. This helps analyze the challenges and identify the research problems that form the core focus of this thesis. We then briefly discuss the proposed solutions to the identified research problems, representing our contributions to the field. In Chapter 2, we first comprehensively outline the existing notions of group and individual fairness and categorize the current algorithms. The chapter further discusses the advantages and disadvantages of existing algorithms in terms of theoretical guarantees, time complexity, and reproducibility. Following this broad discussion, we zoom in on the key focus of our thesis, which is fair algorithms for clustering and recommender systems. The next five chapters (Chapters 3, 4, 5, 6, and 7) address our research problems and present their corresponding solutions. Chapter 8 concludes the thesis by exploring new directions and open challenges in fair clustering and recommender systems. By identifying areas needing further investigation and development, we aim to contribute to ongoing progress in the domain and inspire future researchers to expand upon our work.

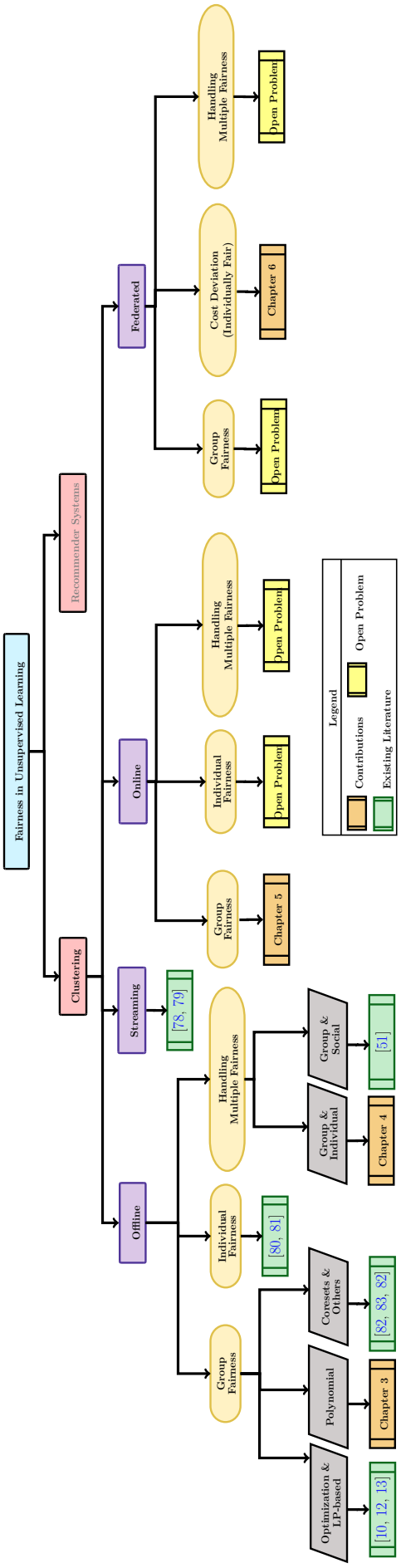


Figure 1.8: A taxonomy of contributions in fair algorithms for clustering.

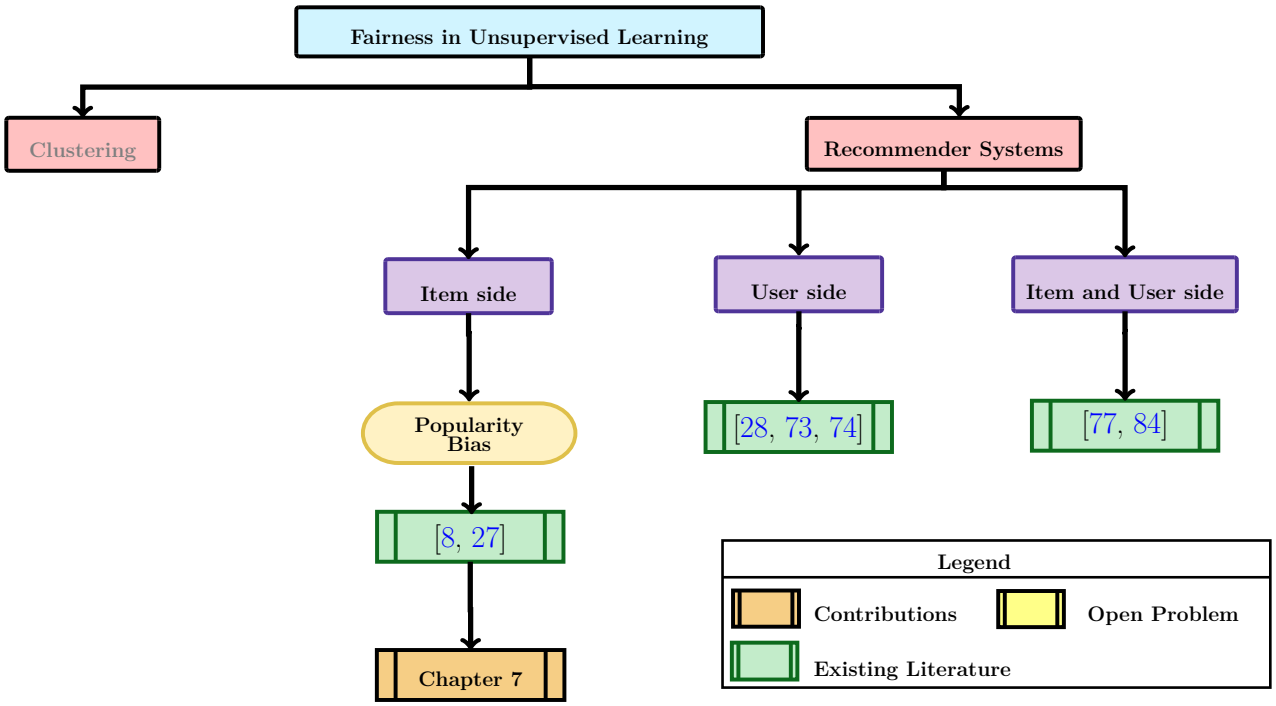


Figure 1.9: A taxonomy of contributions in recommender systems.

Chapter 2

Background

2.1 Clustering

Let $X \subseteq \mathbb{R}^h$ be a finite set of data points that need to be partitioned into k clusters. Each data point $x_i \in X$ is a feature vector described using h real-valued features. A k-clustering[1] $\mathcal{C} = (C, \phi)$ produces a partition of X into k subsets indexed by $[k] = \{1, 2, \ldots, k\}$. The clustering $(\mathcal{C})$ is characterized by a set of centers $C = \{c_j\}_{j=1}^{k}$, an assignment function $\phi : X \to C$ that maps each data point to the corresponding cluster center forming clusters $\{C_1, C_2, \ldots, C_k\}$ respectively. Furthermore, let $d : X \times X \to \mathbb{R}^+ \cup \{0\}$ denote a distance function obeying triangular inequality that measures the dissimilarity between features. Let $\mathbb{I}(\cdot)$ denote the indicator function, which takes a value of one if the condition inside the function is obeyed; otherwise, results in zero. A vanilla (an unconstrained) clustering algorithm determines the cluster centers to minimize the following objective cost:

Definition 2.1 (Objective Cost)

Given p, the clustering objective cost with respect to the metric space (X, d) is defined as:

$$L_p(X, \phi) = \left(\sum_{x_i \in X} d(x_i, \phi(x_i))^p \right)^{\frac{1}{p}} \tag{2.1}$$

Different values of p result in objective cost for different clustering methods i.e., $p = 1$ for k-median, $p = 2$ for k-means, and $p = \infty$ (infinity) for k-center problem. Our aim in this thesis is to develop an algorithm that minimizes the objective cost irrespective of p value while ensuring fairness. Note that in standard vanilla k-means and k-median, objective cost involves a *sum of distances* of data points to the corresponding center with the value of p, as one or two respectively. However, when p takes an infinite value (i.e., k-center problem), the algorithm minimizes the maximum distance of any data point to its center. We now mathematically formulate our **optimization problem** at hand for, say, k-means objective with $z_{i,j}$ as binary variable [2] deciding whether $x_i \in X$ gets assigned to cluster

Some parts of this chapter are accepted as a book chapter in Springer's Ethics in Artificial Intelligence: Bias, Fairness and Beyond book [43].

[1] Throughout the thesis, for simplicity, we call a k-clustering as simply a clustering.

[2] It can also be considered as a variable with the range as real values between 0 and 1 (both inclusive). However, it will then require rounding techniques for computing hard assignments.

center given by assignment function ϕ and y_j as a variable indicating if c_j is opened up as cluster center where $j \in [k]$. Therefore, our problem is as follows:

$$\min_{z_{i,j}, Y_j} \left(\sum_{x_i \in X} (d(x_i, \phi(x_i)))^p \cdot z_{i,j} \right)^{1/p} \tag{2.2}$$

such that

$$\sum_{j \in [k]} z_{i,j} = 1 \quad \forall x_i \in X \tag{2.3}$$

$$z_{i,j} \leq y_j \quad \forall j \in [k], \forall x_i \in X \tag{2.4}$$

$$\sum_{j \in [k]} y_j = k \tag{2.5}$$

In the above optimization problem, Equation 2.2 corresponds to the objective of minimizing objective cost (Definition 2.1). The constraint in Equation 2.3 ensures that each data point is assigned to exactly one cluster center (as $Z_{i,j}$ is binary). The constraint provided in Equation 2.5 and 2.4 ensures that exactly k centers are opened. The above optimization is proved to be NP-hard [32, 33, 35, 85]. Despite NP-hardness, many heuristics and approximation algorithms exist and are widely used in real-world applications [30]. Let the cost approximation factor for such vanilla clustering algorithms be denoted by β. Then the best-known approximation factor (β) values for k-means [86], k-median [33] and k-center [35] objectives are $2, (1 + \sqrt{3} + \epsilon)$ and 2 respectively for small constant $\epsilon > 0$. The above-discussed optimization problem (and referenced approximation or heuristic methods) does not inherently consider any fairness constraints. To formulate such fairness constraints mathematically, we now define different notions of fairness proposed in the past literature.

2.1.1 Fairness in Offline Clustering

We primarily focus on group and individual fairness levels as part of this thesis. Recent works have developed mathematical formulations (known as fairness notions) for handling group and individual fairness in clustering, which are discussed below.

Group Fairness and Notions

The prevalence of anthropological factors such as discrimination based on gender, race, and ethnicity in the data has resulted in a study of group level fairness. Group fairness demands that different protected group values (say male and female for protected group gender) should be treated in an unbiased manner. It is important to note that the protected groups are not restricted to social aspects such as gender but extend beyond to factors such as income levels, education levels, and languages spoken. Moreover, some groups, such as race, can take more than two distinct values (e.g., American, African, Asian), forming **multi-valued protected groups**. Throughout this thesis, let us consider that each data

point, $x_i \in X$ is associated with a *single* protected group $\rho(x_i)$ (say ethnicity from a pool of other available protected groups) that takes values from the set of m values denoted by $[m]$. The number of distinct protected attribute values is finite and much smaller than the size of the dataset[3] (X). Note that the protected group usually corresponds to disadvantaged groups and is known apriori to the algorithms as an additional input. Also, let us denote X_ℓ and n_ℓ as set and number of data points, respectively, having protected group value $\ell \in [m]$ in X. Most current literature focuses on achieving group fairness against a single protected group, aka non-overlapping group identities. However, in practice, the group identities often overlap. For example, a person can belong to two protected groups: race as black and gender as female. Overlapping identities are the focus of a few recent developments [12, 13] (discussed in detail later in this chapter). The thesis will focus on non-overlapping identities.

We first define the notion of group fairness called Balance was proposed for binary protected groups by Chierichetti et al. [42] and extended to the multi-valued groups by Bera et al. [12] and Ziko et al. [10]. The balanced fairness notion is defined as follows.

Definition 2.2 (τ-Balance)

For a binary valued protected group taking values from set $\{\ell_1, \ell_2\}$, a clustering $\mathcal{C}$ is said to be τ-Balance [42] with

$$\tau = \min_{C_j \in \mathcal{C}} \left(\min \left(\frac{\sum_{x_i \in C_j} \mathbb{I}(\rho(x_i) = \ell_1)}{\sum_{x_i \in C_j} \mathbb{I}(\rho(x_i) = \ell_2)}, \frac{\sum_{x_i \in C_j} \mathbb{I}(\rho(x_i) = \ell_2)}{\sum_{x_i \in C_j} \mathbb{I}(\rho(x_i) = \ell_1)} \right) \right). \tag{2.6}$$

Balance is computed by finding the minimum possible ratio of protected (say, male) and non-protected group (say, female) over all clusters. Any fair clustering algorithm using Balance as a measure of fairness would produce clusters that maximize the τ value. It is easy to see that the maximum value of τ in τ-Balance is equal to the *dataset ratio*, i.e., the setting when each cluster receives data points in the same fraction as that present in the dataset. This is supported by the fact that if one tries to improve the balance of a cluster beyond this limit, then it will lead to the degradation of the balance of some other clusters, resulting in a decrease in the overall balance of the clustering. It is important to note that Balance notion does not allow the user to provide a trade-off between the clustering objective and fairness. Further, the clusters maximizing the Balance are not unique. Let us take an example to understand the notion with the binary-protected group taking two values, red and blue. A clustering algorithm divides the data points into two clusters with 12 red and 3 blue data points in one cluster; and 3 red and 3 blue in another cluster. Such a clustering is said to obey 0.25-Balance i.e. $\min\left(\min(\frac{12}{3}, \frac{3}{12}), \min(\frac{3}{3}, \frac{3}{3})\right)$. The dataset ratio, however, is $\frac{6}{15} = 0.4$, and as can be seen, red data points significantly dominate blue data points in the first cluster. A more balanced clustering would be with 7 red, 3 blue data points in the first cluster and 8 red, 3 blue data points in the other

[3]Otherwise, the problem is uninteresting as the balanced clustering may not be feasible.

cluster.

However, τ-BALANCE is restricted to binary protected groups. More generic fairness notions, i.e., Restricted dominance ($\boldsymbol{\tau}$-RD) and Minority protection ($\boldsymbol{\tau}$-MP), avoid dominance and preserve the minimal representation of a single group, respectively are defined below:

Definition 2.3 (τ-RD)

A clustering $\mathcal{C}$ is said to obey **restricted dominance** with respect to $\boldsymbol{\tau}$ (i.e. $\boldsymbol{\tau}$-RD [12]) if for all $\ell \in [m], C_j \in \mathcal{C}$,

$$\sum_{x_i \in C_j} \mathbb{I}(\rho(x_i) = \ell) \leq \tau_\ell |C_j|. \tag{2.7}$$

Definition 2.4 (τ-MP)

A clustering $\mathcal{C}$ is said to obey **minority protection** with respect to $\boldsymbol{\tau}$ (i.e. $\boldsymbol{\tau}$-MP [12]) if for all $\ell \in [m], C_j \in \mathcal{C}$,

$$\sum_{x_i \in C_j} \mathbb{I}(\rho(x_i) = \ell) \geq \tau_\ell |C_j|. \tag{2.8}$$

In the same example as above, if we consider notions of $\boldsymbol{\tau}$-MP and $\boldsymbol{\tau}$-RD, then the clustering satisfies $(0.2, 0.5)$-MP and $(0.8, 0.5)$-RD[4]. Note that all the existing notions further determine fairness either using cluster sizes, which are unknown apriori to the algorithm, or are limited to binary protected group values. Next, we define the τ-FE notion. It also considers fairness with respect to the number of data points in each cluster but leads to a continuous and convex optimization objective. Note that f-divergence can also be used instead of KL-divergence [64] in τ-FE.

Definition 2.5 (τ-FE)

The fairness error (τ-FE [10]) of a clustering $\mathcal{C}$ with respect to a given vector $\boldsymbol{\tau}$ is defined as:

$$\sum_{j \in [k]} \mathcal{D}_{KL}(\boldsymbol{\tau} || P_j) = \sum_{j \in [k]} \sum_{\ell \in [m]} -\tau_\ell \log P_j^\ell \tag{2.9}$$

where, $\mathcal{D}_{KL}$ is the Kullback-Leibler (KL) divergence and P_j^ℓ is the fraction of data points with protected group value ℓ in cluster j.

Individual Fairness and Notions

Group fairness does not ensure fair treatment for a particular individual. The trait of human envy might still make an individual discontented. For example, an employee might feel discriminated against or left out if similar employees receive a favorable appraisal. There are algorithms in the literature that guarantee individual fairness

[4] $\boldsymbol{\tau}$ vector is written in the form (red, blue) respectively in $\boldsymbol{\tau}$-MP, $\boldsymbol{\tau}$-RD notion.

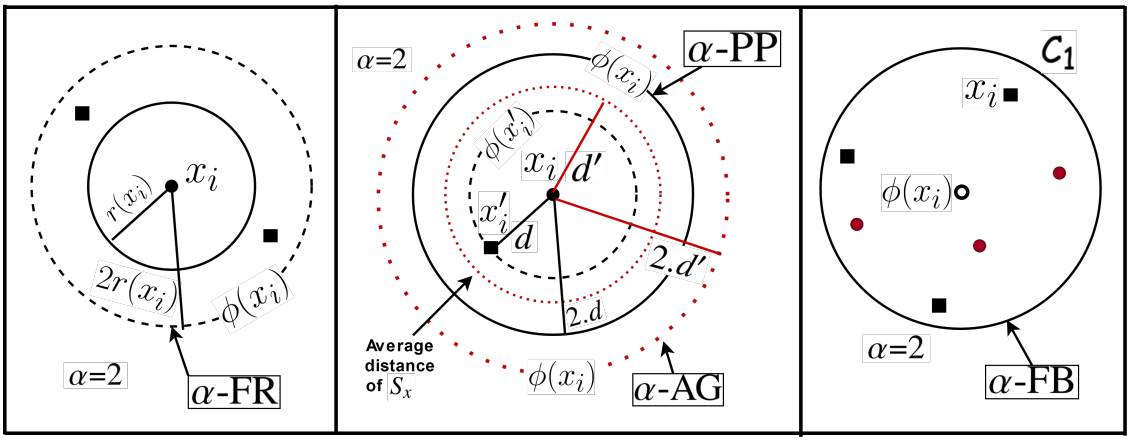


Figure 2.1: Individually fair notions: (a) Given a data point x_i, 2-FR demands that center for x_i (denoted by $\phi(x_i)$) lies at most within $2r(x_i)$ from x (dotted line). (b) 2-PP suggests the center be within twice the minimum center distance of a data point, say x'_i in similarity set $S(x_i)$, i.e., within $2d$. 2-AG relaxes the distance to $2d'$ by taking the average distance d'. (c) 2-FB demands at least two data points of similar type in the cluster.

[80, 14]. ***Individual level fairness*** is not tied to protected groups but rather to 'similar' individuals. Let each data point x_i identify itself with other similar data points represented by the set $S(x_i)$. Further, let $r(x_i)$ be the minimum radius of a ball $\mathcal{B}(x_i, r(x_i))$ centered around x_i that contains n/k points, where k is the number of clusters.

The fundamental principle behind individual fairness is that similar individuals expect similar treatment. Any deviation would induce an unfair feeling in an individual [87, 88]. Various notions of individual fairness differ in how individuals perceive similarity and are discussed below:

Definition 2.6 (α-FR Fairness)

A clustering $\mathcal{C}$ is said to be α-FR fair [80] if for $\alpha \geq 0$, $\mathcal{C}$ obeys

$$d(x_i, \phi(x_i)) \leq \alpha r(x_i) \quad \forall x_i \in X. \tag{2.10}$$

The α-FR notion assures that any data point x_i has its center within a radius containing n/k neighbours of x_i. The rationale behind n/k is that every center, on expectation, assigned n/k data points. Note that the performance of individually fair algorithms approximating α-FR is measured in terms of the value of α. This notion is restrictive as the neighbours of x_i are also determined using distance function $d(\cdot)$. We now define more generalized notions.

Definition 2.7 (α-PP Equitable Fairness)

(Per point Fairness) [89] A clustering $\mathcal{C}$ is said to be α-PP fair if for $\alpha \geq 0$, $\mathcal{C}$ obeys

$$d(x_i, \phi(x_i)) \leq \alpha \left(\min_{x'_i \in S(x_i)} d(x'_i, \phi(x'_i)) \right) \quad \forall x_i \in X \tag{2.11}$$

Definition 2.8 (α-Ag Equitable)

(Aggregate Fairness) [89] A clustering $\mathcal{C}$ is said to be α-AG fair if for $\alpha \geq 0$, $\mathcal{C}$ obeys

$$d(x_i, \phi(x_i)) \leq \alpha \left(\frac{\sum_{x_i' \in S(x_i)} d(x_i', \phi(x_i'))}{|S(x_i)|} \right) \quad \forall x_i \in X. \tag{2.12}$$

Definition 2.9 (α-FB Fairness)

(Feature based) [14] A clustering $\mathcal{C}$ is said to be α-FB fair if for $\alpha \geq 0$, and similarity set $S(x_i)$, $\mathcal{C}$ obeys

$$|x_i' \in S(x_i) \text{ and } \phi(x_i) = \phi(x_i')| \geq \alpha \quad \forall x_i \in X. \tag{2.13}$$

In contrast to α-FR notion, the individual fairness notions, namely α-PP, α-AG, and α-FB allow for an explicit similarity set $S(x_i)$ (perhaps determined through distance or number of matching features). These three notions propound the idea that similar data points should be clustered similarly. We summarize all these individual fairness notions in Figure 2.1. Next, we define the Avg-dist notion, which uses well-known clustering stability ideas [87] and the game-theoretic concept of average attraction properties [90]. It induces the individual fairness notion that data point x_i should be closer to its own cluster members than data points from other clusters.

Definition 2.10 (Avg-dist Notion)

(Kleindessner et al. [91]) A clustering $\mathcal{C}$ is said to obey Avg-dist Notion if $\forall x \in C_j$,

$$\frac{1}{|C_j| - 1} \sum_{y \in C_j / x} d(x, y) \leq \frac{1}{|C_i|} \sum_{y \in C_i} d(x, y) \quad \forall i \neq j \in [k]. \tag{2.14}$$

Having described existing group and individual fairness notions, We now provide a detailed survey of the present state-of-the-art algorithms to handle fairness in offline clustering.

Taxonomy of Algorithms

Typical fair clustering solutions aim to minimize the standard clustering objectives while simultaneously enforcing fairness. The stage (*pre-processing, in-processing, and post-processing*) at which fairness constraints are enforced is a key differentiating factor of the existing offline algorithms. Pre-processing techniques alleviate data bias before clustering by adding restrictions on the distribution of data points among clusters by a clustering algorithm [42, 11]. In contrast, in-processing techniques intertwine the clustering and fairness imposition parts of the algorithm [44, 92]. Finally, fairness is an afterthought for post-processing interventions that typically redistribute the instances of clusters obtained by vanilla clustering to obtain fair clusters [12, 13].

The solution framework is another distinguishing factor for fair clustering techniques. While some approaches add a regularizer term encoding the fairness to the clustering

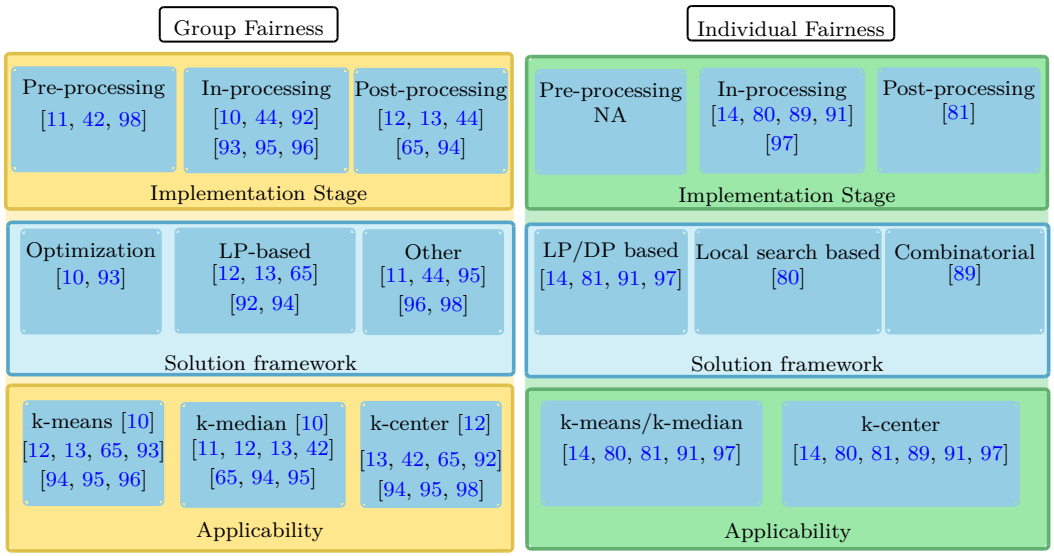


Figure 2.2: Taxonomy of group and individual fairness in clustering algorithms.

objective cost [10, 93], other approaches propose linear programming (LP) formulation of the fair clustering problem with linear fairness constraints [12, 13, 94, 92]. Other group fair clustering solutions include tree-based structures, round-robin allocation, and decision problems (such as min-cost flow and perfect matching algorithms) [44, 42, 11, 95, 96]. Solution approaches to individual fairness clustering include LP-based formulations [81, 14], local search approaches [80], dynamic programming [91], and combinatorial optimization [97]. Note that all the existing algorithms apply only to certain clustering objectives (k-means/k-median/k-center). So, the applicability of the approach is another differentiating factor. Figure 2.2 shows the taxonomy of existing algorithms categorized along (i) different stages of implementation, (ii) underlying solution frameworks, and (iii) applicability.

Many state-of-the-art (`SOTA`) techniques are supported by theoretical guarantees on fairness and the quality of the clusters, along with detailed cost approximation and computational complexity analysis. With the growing number of new fairness notions and algorithms, we now comprehensively review the methodology, theoretical underpinnings and computational challenges of both group and individual fair offline algorithms. We also discuss various advantages and disadvantages of each of the approaches. This will help identify the successes and future directions for the research community to work upon.

Algorithmic Details and Theoretical Guarantees

Group Fairness:

The foundational work of Chierichetti et al. [42] partitions the data points into small clusters, namely *fairlets*. The paper shows that finding optimal fairlet decomposition is NP-Hard. To find approximate fairlet decomposition, authors use the strategy of solving bipartite matching [99] for maximally balanced clustering (i.e., achieving balance equal to dataset ratio) and minimum cost flow instances otherwise [100].

The fairlets formed are then merged into k clusters by applying standard (or vanilla) clustering (k-center/k-median) on fairlet centers. The algorithms achieve 4-approximation guarantees on cost with satisfying τ-Balance for k-center and a $(\frac{1}{\tau} + 1 + \sqrt{3} + \epsilon)$-approximation cost guarantee for the k-median objective; where, ϵ is a positive constant. The following are three major shortcomings of this approach: 1) It works only for binary-valued protected groups, 2) It can only achieve the Balance same as the n data points in $X \subseteq \mathbb{R}^h$ i.e., dataset ratio, and 3) It is not scalable for large datasets. To make the approach scalable, Backurs et al. [11] proposed a near-linear time algorithm to compute fairlets using QuadTree data structure [101]. The algorithm computes an embedding with k-median cost approximation of $O(hlog(n))$. However, this work is limited to binary-valued protected groups with k-median clustering objective and can only achieve *dataset ratio*.

Extension to multi-valued protected groups is considered in Böhm et al. [95], which proposes a minimum cost-perfect matching (MCPM) algorithm. They provide algorithms for k-center and k-median clustering objectives with 3-approximation and $(\beta + 2)$-approximation[5] fairness guarantee on τ-Balance. However, the algorithm works only when the number of data points from each protected group is equal in the dataset. A similar 14-approximation approach using MCPM is proposed in Rösner and Schmidt [98] for τ-Balance. They further propose a 4-approximation method for τ-MP fairness using a reduction to the maximum flow problem. The work is limited to the k-center model.

Among the LP-based techniques, Bera et al. [12] formulate fair clustering as a linear program with $\boldsymbol{\tau}$-RD and $\boldsymbol{\tau}$-MP as constraints. This paper guarantees a maximum fairness violation of at most 3 while simultaneously satisfying $(\beta + 2)$-approximation guarantee on the objective cost. Harb et al. [13] extend these guarantees for fair k-center for multi-valued protected groups by formulating LP by restricting the search space for better time complexity. The work by Ahmadian et al. [92] solves fair clustering via $\boldsymbol{\tau}$-RD along with an additional constraint on representative fairness [61]. The authors prove that it is NP-hard to obtain an algorithm better than 2-approximation for $\boldsymbol{\tau}$-RD $\in (0, 0.5]$. The proposed algorithm with maximum $O(n^2)$ constraints and variables is 3-approximation while the case ($\boldsymbol{\tau}$-RD=0.5) with $O(nk)$ constraints and variables is 12-approximation in the clustering objective. The work by Bercea et al. [102] also proposes an LP formulation for $\boldsymbol{\tau}$-MP and $\boldsymbol{\tau}$-RD fairness. The approach achieves a $3, 4.675$, and 62.856-approximation for k-center, k-median, and k-means, respectively.

Several other works do not provide any theoretical guarantee on the quality of fair clusters. The work by Davidson et al. [65] propose a post-processing technique to impose fairness (in terms of $\boldsymbol{\tau}$-RD and $\boldsymbol{\tau}$-MP) after cluster formation by assigning points from protected groups equally among all k clusters. The goal is to have fewer disagreements between solutions. This approach uses integer linear program (ILP) formulation and uses *total unimodularity* structure of the problem to bypass computational intractability. The

[5]To recall, β is the approximation factor of vanilla clustering algorithms.

	Fairness Notions	**Time Complexity**			**Cost Approximation Factor**		
		***k*-means**	***k*-median**	***k*-center**	***k*-means**	***k*-median**	***k*-center**
[93]	FairKM	$O(n^2hk)$	×		×		
[92]	$\boldsymbol{\tau}$-RD	×		Max. variables, constraints: n^2 & nk for τ_ℓ=0.5 $\forall\ \ell \in [m]$	×		3 & 12 for τ_ℓ=0.5 $\forall \ell \in [m]$
[11]*	τ-BALANCE	×	$O(hn\log(n))$	×	×	$O(hlog(n))$	×
[12]*	τ-BALANCE $\boldsymbol{\tau}$-RD & $\boldsymbol{\tau}$-MP	Max. variables & constraints: $O(n^2)$			$(\rho+2)$		
[102]	$\boldsymbol{\tau}$-RD & $\boldsymbol{\tau}$-MP	×			3	4.675	62.856
[95]	τ-BALANCE	$O(n^3T)$	$O(nh)$	$O(nhk)$	$(\beta+2)$		3
[42]*	τ-BALANCE	×	$O(T+n^2)$		×	$(\tau+1+\sqrt{3}+\epsilon)$	4
[65]	$\boldsymbol{\tau}$-RD & $\boldsymbol{\tau}$-MP	nk regular variables , $2k$ slack variables & $2k+n$ constraints			×		
[94]	$\boldsymbol{\tau}$-RD & $\boldsymbol{\tau}$-MP	Variables & constraints: $O(n^2)$			×		
[44]*	τ-BALANCE τ-FE	$O(kn\log n)$		×	$2(\beta+2)$		×
[13]*	$\boldsymbol{\tau}$-RD & $\boldsymbol{\tau}$-MP	Max. variables: $\min(2^{k-1}k\mid I\mid, nk)$ & Max.constraints: $km+\min(2^k\mid I\mid, nk)$			×		
[96]*	τ-BALANCE	×			×		
[98]	τ-BALANCE τMP	×		Polynomial	×		14 for τ-BALANCE, 4 for τ-MP
[10]*	τ-BALANCE τ-FE	$O(n^2k^2h)$		×	×		

Table 2.1: Categorization of group fairness clustering algorithms. The variable $|I| \leq n$ in [13] and T is time taken by vanilla clustering. (*source code is available and well tested by us).

complementary problem of minimizing unfairness with maximum allowable clustering cost is considered in Esmaeili et al. [94] using LP formulation. To bound the number of LP iterations, the algorithm exhaustively searches for the feasibility of LPs and chooses the solution with minimum fairness constraints. The integral solution is then constructed using a network flow [102].

Among the regularized-based techniques, Ziko et al. [10] propose a *variational* framework where clustering and fairness objectives are simultaneously solved as an optimization problem. This paper uses $\boldsymbol{\tau}$-FE as the fairness notion and breaks the composite problem into convex and concave parts, which are bounded by auxiliary functions. These functions help compute the soft assignment update in the subsequent iteration of k-means, k-median, and N-cut [103]. The authors show that the variational framework has monotonicity and convergence guarantees as Expectation-Maximization (EM) algorithms [104]. The main issue with the approach is the use of data-dependent hyper-parameters. Another important limitation of this approach is that clustering objective cost deteriorates significantly with an increase in the number of clusters (Chapter 3). Liu et al. [96] formulate the problem of fair clustering as a bi-objective optimization problem with τ-BALANCE notion of fairness and prove a sublinear convergence rate. The resulting objective function is non-convex; hence, the solution obtained by stochastic gradient descent does not satisfy any theoretical

guarantees on the quality of obtained clusters. Table 2.1 summarizes all the results.

Extension of Group Fair Algorithms to Multiple Protected Attributes: Group fairness constraints are also studied under *multiple* multi-valued protected groups setting. For example, an individual can be a female (gender) and native-American (ethnicity). In clustering, this overlap between multiple protected groups is denoted by Δ(=2 in the above example). Both $\boldsymbol{\tau}$-RD and $\boldsymbol{\tau}$-MP can be extended to multiple mulit-valued protected groups. The work by Bera et al. [12] is also applicable to multiple protected groups with maximum additive violation of $4\Delta + 3$ for $\Delta \geq 2$ (+3 for Δ=1). The work by Harb and Lam [13] provides a similar guarantee.

A notion similar to $\boldsymbol{\tau}$-FE is proposed by Abraham et al. [93] for multiple protected groups. The authors propose a fair k-means algorithm (FairKM) for solving a combined objective function of minimizing objective cost along with a deviation in this modified notion of fairness. The algorithm, however, is sensitive to the trade-off parameter, needs extensive tuning, and requires minimization of a non-convex function

Individual Fairness

We now discuss the algorithmic framework and theoretical guarantees of individually fair clustering algorithms. To satisfy α-FR fairness guarantee, a set of *critical balls* is determined [80, 81, 97]. Each critical ball contains a set of data points and a *critical* center with the property that each data point in a critical ball has a distance less than a pre-defined value from the critical center. In Mahabadi and Vakilian [80], the critical balls are defined such that all the data points have distance within $6\alpha r$ to the critical center; here r is defined as the minimum radius containing n/k data points from any data point. These critical balls are identified using the modified version of greedy approaches proposed in [105, 106]. Next, they use a local search algorithm to improve clustering objective cost and achieve a bicriteria approximation guarantee[6] of $(84, 7)$-approximation for α-FR and $(O(p), 7)$-approximation for general p-norm with k-median as clustering objective.

On similar lines, Vakilian and Yalçıner [97] consider critical balls of radius $2\alpha r$. For fair k-median, authors use k-median algorithm by Swamy [107], and for k-center a reduction to standard k-center problem is presented that achieves $(8 + \epsilon, 3)$-approximation solution. For general $p > 1$, a $(16^p, 3)$-approximation reduction to matroid facility location problem solved using LP relaxation is proposed. The approximation guarantee is further improved to $(8, 2^{(1+2/p)})$-approximation by Negahbani et al. [81], who proposed a fair rounding technique to the optimal LP solution computed using critical centers with radius $2r$.

A feasible solution is not guaranteed for α-PP and α-AG with $\alpha < 2$ [89]. However, any instance with $\alpha \geq 2$ always admits a feasible solution. Even with $\alpha \geq 2$, authors provide an instance where the *price of fairness* [7] without any additional constraint can be arbitrarily bad. For finding feasible centers and fair assignments, the authors provide an algorithm having 5-approximation on the fairness guarantee.

[6] (p, q)-approximation bicriteria denotes cost approximation of p and fairness approximation of q.

[7] Ratio of clustering objective value under fairness constraint to the unfair (standard) objective value.

	Fairness Notion	**Time Complexity**	**Cost Approximation Factor**		
			k**-means**	k**-median**	k**-center**
[89]*	α-PP α-AG	×	5-approximation w.r.t fairness		
[14]	α-FB	Polynomial	$((1+\delta)\text{OPT}(\beta+2))$		
[91]	Avg. dist based	$O(n^3k)$	×		
[80]*	α-FR	$O(k^5n^4)$	$(O(p), 7)$	$(84, 7)$	$(O(p), 7)$
[81]*	α-FR	$O(kn^4)$	$(8, 2^{1+\frac{2}{p}})$		
[97]	α-FR	Polynomial	$(16^p, 3)$	$(8+\epsilon, 3)$	$(8+\epsilon, 3)$

Table 2.2: Categorization of individual fair clustering algorithms. $\mathcal{OPT}$ is optimal for fair assignment cost in [14].(*source code is available and well tested by us).

Kar et al. [14] show that finding α-FB fair clustering is NP-complete even for $k = 2$. The authors provide a $(1+\epsilon)\text{OPT}(\beta+2)$-approximation randomized algorithm solved with the help of LP-relaxation for fair assignment where OPT is optimal fair assignment cost. Similar to α-FB, finding a clustering satisfying *Avg-dist* fairness notion is proved to be NP-Hard even for dataset $X \subseteq \mathbb{R}^2$ ([91]). The authors in [91] further present a dynamic programming-based solution 1-dimensional setting to find contiguous clusters of target sizes. Table 2.2 summarizes all the results for individually fair algorithms.

2.1.2 Fairness in Online Clustering

A more stringent variation of offline and streaming environments is online clustering, where an endless stream of data points arrives over time. Let $X \subseteq \mathbb{R}^h$ be an endless stream of data points with x_t being the point arriving at time t. Each data point $x_t \in X$ is articulated using h dimensional real-valued features. Due to limited memory, the algorithm must make an irrevocable decision about incorporating an incoming data point into existing clusters or opening it as a new center. Once a data point becomes a center, it remains so forever. Similarly, any data point previously seen cannot be chosen as the center when a new data point arrives [38, 39]. An important aspect to note in online clustering pertains to the absence of information regarding the ordering of the arrival of data points in the stream. As a result, the algorithm ends up opening more number of centers ($k_{\texttt{actual}}$) than the desired target ($k_{\texttt{target}}$), i.e., $k_{\texttt{actual}} \geq k_{\texttt{target}}$ to maintain good approximation guarantees on objective cost. Note that $k_{\texttt{target}}$ and k are used interchangeably for ease of reading. Further, all other notations remain intact as offline clustering.

Group Fairness and Online Algorithms

In an offline setup, imposing a minimum threshold of data points from each group value in every cluster is feasible as the number of clusters (k) and total number of data points (n) are fixed. However, in an online setting, n is not restricted, and the number of centers opening up is not fixed; therefore, imposing a lower bound on the number of data points from each group value is less practical for maintaining fairness. There are high odds that

the data points belonging to non-protected group value may eventually start dominating over time in a cluster due to an endless stream of data points. Thus, there is a need to devise a notion of group fairness for online setup, and there is no present work that handles group fairness in online clustering to the best of our knowledge. We will address this open direction as a part of this thesis (Chapter 5).

2.1.3 Fairness in Federated Data Clustering

For federated settings, we retain all notations but redefine a few notations, making them separate for clients and servers. Therefore, let $X \subseteq \mathbb{R}^h$ be a set of data points distributed among Z clients. Let the data points on any client $z \in [Z]$ be $X^{(z)}$. Each data point in $X^{(z)}$ is again a h-dimensional real-valued feature vector. Note that the complete set of data points X contain data points belonging to $[k]$ different true distributions. The goal of any clustering algorithm is to partition the data points spread across clients into a set of disjoint sets (called clusters) represented by the set of global centers denoted by set $C^g = \{c_1^g, c_2^g, \ldots, c_k^g\}$. The computation of finding these global centers involves initially computing the best local centers that partition the local data $X^{(z)}$ (for any $z \in [Z]$) into k disjoint sets represented by $C^{(z)} = \{c_1^{(z)}, c_2^{(z)}, \ldots, c_k^{(z)}\}$. We denote the local assignment function at each client over any center set (say $C^{(z)}$) by $\phi^{(z)} : X^{(z)} \to C^{(z)}$. Note that the data points on any client z may not belong to all $[k]$ distributions, and this idea is captured using the notion of heterogeneity in federated settings. Formally, it is defined as follows:

Definition 2.11 (Heterogeneity)

Given k, the heterogeneity level (denoted by $\mathtt{H}$) determines the maximum number of distributions the data points $X^{(z)}$ on a client $z \in [Z]$ belongs to, i.e., $\mathtt{H} \leq k$.

In practice, determining the exact level of heterogeneity ($\mathtt{H}$) on a client is often not feasible. Consequently, a common approach in federated data clustering literature is to compute k ($\geq \mathtt{H}$) partitions on each client [41, 108]. These partitions are not arbitrary selections but are the one that minimizes the following objective cost using final converged global centers:

Definition 2.12 (Objective Cost)

Given k, $\cup_{z\in[Z]} X^{(z)}$, and distance metric $d : X \times X \to \mathbb{R}^+ \cup \{0\}$ with norm value p the local objective cost $L_p^{(z)}$ of client z of (k, p)-clustering in a federated setting with a set of centers C is computed as follows:

$$L_p^{(z)}(C) = \left(\sum_{x_i \in X^{(z)}} \left(d(x_i, \phi_C^{(z)}(x_i)) \right)^p \right)^{1/p} \tag{2.15}$$

In a federated setup, comparing methods based on the **mean objective cost per data point** is often more realistic than the total objective cost at a client. The primary reason

is that dataset sizes across clients can differ significantly in federated settings. Thus, evaluating the per-point cost incurred by clients makes more sense. Mathematically, this can be formulated as follows:

$$\boldsymbol{\mu}^{(z)}(C^g) = \frac{L_p^{(z)}(C^g)}{|X^{(z)}|} \tag{2.16}$$

where $\boldsymbol{\mu}^{(z)}(C^g)$ is the mean cost per data point on any client z.

Fairness Notion for Federated Data Clustering

In a federated setting, the objective cost suffered by any client z can significantly differ from that of other clients because data points from different ($\mathtt{H} \leq k$) distributions can be distributed (or generated) in a highly skewed manner among (or at) clients. Therefore, if the global centers deviate too much from the best local centers, clients might feel reluctant to contribute to the federated environment to learn a better global center representation. Thus, the aim is to not solely focus on minimizing global objective cost (or per point cost) but rather to find a k-clustering in the federated setting that is fair for all clients, i.e., one which achieves near uniform cost across all clients. We formally define such a clustering as follows:

Definition 2.13 (Fair Federated Data Clustering)

Given that data points are sampled from k true clusters and are distributed over Z clients. Then, for any two set of federated global centers C_1^g and C_2^g, we say that C_1^g is more fair than C_2^g if the **cost deviation per data point** (σ) is lower for C_1^g than C_2^g. Here $\boldsymbol{\sigma}$ over centers C_i^g for $i \in \{1, 2\}$ is given as follows:

$$\boldsymbol{\sigma}(C_i^g) = \sqrt{\frac{\sum_{z \in [Z]} \left(\boldsymbol{\mu}^{(z)}(C_i^g) - \boldsymbol{\mu}(C_i^g)\right)^2}{Z}} \tag{2.17}$$

Note that here $\boldsymbol{\mu}(C^g)$ is the mean value of $\boldsymbol{\mu}^{(z)}(C^g)$ across all clients.

The notion captures the idea analogous to individual fairness and demands that the federated clustering model should treat all clients similarly i.e., all clients should face similar clustering objective cost.

Algorithms for Federated Data Clustering

Dennis et al. [41] makes an initial effort to partition the data points and proposes an algorithm which they call `k-FED`. The algorithm builds upon the Awasthi and Sheffet [109] aka (Awasthi), assuming the centers are well separated and clusters follow gaussian distribution properties. `k-FED` executes Awasthi locally on each device to find k local centers, which are then communicated to the server for computing the final clustering. The server then employs a farthest heuristic similar to the offline k-center approach[8].

[8] https://cseweb.ucsd.edu/ dasgupta/291-unsup/

Yang et al. [110] proposes a slightly enhanced greedy centroid-based initialization for `k-FED` which surpasses centralized k-means in specific scenarios.

Some works in this direction approach the problem by framing it as a generative data synthesis challenge and leveraging concepts from Generative Adversarial Networks (GANs) [111, 40, 112, 113, 114]. The broader picture involves training multiple GANs locally at clients and utilizing their parameters to construct a global GAN model. This global GAN model is employed to generate synthetic data and further identify k distinct cluster centers. These centers are subsequently communicated back to clients to partition their local data points. Li et al. [115] also pursues a parallel approach to develop privacy-preserving distributed clustering by incorporating concepts from cryptography. The proposed method initially computes local center updates and then shares encrypted information using Lagrange encoding back to the server. Thereafter, the server aggregates all secret distance codes from the clients and performs subsequent communication updates. While the algorithm harnesses the advantages of encryption-decryption to safeguard data privacy, such techniques entail substantial computation overhead and communication costs, thereby hindering the scalability of the approach. Similarly, Leeuw [116] employs federated data clustering within the blockchain's committee-based consensus protocol. However, the additional overheads counterbalance the performance improvement.

Fair Federated Data Clustering: It is important to note that no existing work in federated data clustering has specifically focused on addressing the challenge of cost distribution spread across clients and fostering a more equitable clustering as a primary goal. We will address this direction as well in the present thesis (Chapter 6).

2.2 Recommender Systems

Recommender systems are machine learning models that suggest users with items based on their past history or preferences. These preferences are captured either using implicit methods such as click rate or search pattern on items or using explicit methods such as based on ratings. Consider the data with $\mathcal{U}$ denoting the set of users and $\mathcal{I}$ being the set of items. Let $R = \mathcal{U} \times \mathcal{I}$ be a rating matrix where each entry $R_{u,i}$ corresponds to the true rating of item i by the user u on a scale of 1 (lowest) to 5 (highest). All non-interacted user-item (u, i) pairs have a value of $R_{u,i} = 0$. The prediction matrix is given by P with each entry $P_{u,i}$ as the predicted rating for user u and item i. The goal of an ideal recommendation algorithm is to reduce the following loss function:

$$L_{\text{ideal}}(R, P) = \frac{1}{|\mathcal{U}||\mathcal{I}|} \sum_{u \in \mathcal{U}} \sum_{i \in \mathcal{I}} \delta(R_{u,i}, P_{u,i}) \tag{2.18}$$

where the error function δ could be the mean squared error (MSE) or mean absolute error (MAE). Since the true rating $R_{u,i}$ is not available for all possible user-item interactions,

one tends to minimize the loss on the observed set of user-item interactions given by:

$$L_{\text{obs}}(R, P) = \frac{\sum_{u \in \mathcal{U}, i \in \mathcal{I}} \mathbb{I}(R_{u,i} \neq 0)\ \delta(R_{u,i}, P_{u,i})}{\sum_{u \in \mathcal{U}, i \in \mathcal{I}} \mathbb{I}(R_{u,i} \neq 0)} \tag{2.19}$$

Broadly, recommender system algorithms are classified into two techniques - one is content filtering, and the second is collaborative filtering methods. Content filtering techniques rely on creating user-tailored profiles about preferences and tastes. Such user profiles are commonly gathered by querying users with fixed questions, for example, about genre, actors, and audio languages in movie recommendation systems. The system then recommends users by matching items based on user profiles. The main challenge in such methods is capturing external information about preferences. To overcome such challenges, literature proposes collaborative filtering methods that recommend items by analyzing the relationship between users and items with the help of past history, such as ratings. One of the popular collaborative filtering methods of interest for this thesis is Matrix Factorization (MF). We will now look into the mathematical formulation behind Matrix Factorization.

2.2.1 Matrix Factorization

Matrix Factorization is a collaborative filtering method that relies on the idea of latent factors. These factors can be considered an abstraction of different factors or dimensions to understand user-item interactions. For instance, from an item's perspective, such as movies, these latent factors can capture dimensions like level of comedy or romance, orientation to kids, or even can be less well defined, say the depth of suspense or otherwise can be completely uninterpretable dimensions. From the user's perspective, latent factors can capture the scale of favouritism, such as comedy genres and preference for kid content. Now, having a glance through the intuition behind latent factors, we look into the finer details of matrix factorization.

Matrix Factorization mathematically captures both users and items by mapping them as vectors into the latent space of dimensionality κ, The mapping is computed in a way that user-item interactions are captured as the inner product of user vector $\xi_u \in \mathbb{R}^\kappa$ for user u and item vector $\psi_i \in \mathbb{R}^\kappa$ for item i. Intuitively, these vectors capture the level of presence of each of the κ latent factors. Therefore, the predicted rating of user u for item i is, in turn, given by dot product: $\psi_i^T \cdot \xi_u$ i.e., $P_{u,i} = \psi_i^T \cdot \xi_u$. Now, having both predicted and true rating, one can use popular methods such as stochastic gradient descent [117], to minimize the loss function (Equation 2.19) for each interacted user-item (u, i) pairs (that is when $R_{u,i} \neq 0$). The resulting gradient update are provided below in Equations 2.20 and 2.21 for both user vector (ξ_u) and item vector (ψ_i), respectively, with η as the learning rate and $\delta(\cdot)$ as the mean square error:

$$\psi_i^{t+1} \leftarrow \psi_i^t + 2\eta\Big(R_{u,i} - P_{u,i}\Big)\xi_u^t \tag{2.20}$$

$$\xi_u^{t+1} \leftarrow \xi_u^t + 2\eta\Big(R_{u,i} - P_{u,i}\Big)\psi_i^t \tag{2.21}$$

2.2.2 Popularity Bias on Item-side

Recommender systems suffer from fairness issues, such as the presence of popularity bias on both user and item sides, as discussed in Section 1.3.2. We focus on popularity bias on the item side for this thesis. This occurs when popular items (i.e., items with high rating frequency) are recommended more often to users than non-popular items, even if the user has an interest in the latter. In other words, the algorithms that aim solely to minimize L_{obs} can result in recommender systems that are highly accurate for popular items but suffer heavy losses on non-popular items. The primary reason behind this is that inherently popular items are rated more frequently and are available more in the dataset. To this, let $\mathcal{I}_{\mathcal{P}}$ and $\mathcal{I}_{\mathcal{NP}} = \mathcal{I} \setminus \mathcal{I}_{\mathcal{P}}$ denote the set popular items and non-popular items respectively. Inspired by Abdollahpouri et al. [118], we use a threshold mechanism to generate $\mathcal{I}_{\mathcal{P}}$ and $\mathcal{I}_{\mathcal{NP}}$ and obeying Pareto principles [119, 120], we set the threshold as $80:20$. That is, we use top 20% items in terms of rating frequency as popular and remaining as non-popular (long tail) items.

2.2.3 Algorithms for handling Popularity Bias

The adverse effects of popularity bias on users of different demographics are analyzed in Abdollahpouri et al. [118]. A few recent works propose different metrics to evaluate popularity bias. It includes an NDCG metric-based [121] and ARP-based [119, 122] approach. The NDCG metric computes the relevancy of the results and measures the goodness of the ranked ordering of items, whereas ARP computes the average popularity of each item and aims to improve diversity. However, none of these metrics seeks to reduce the disparity between items of different popularity and will be the focus of this thesis (Chapter 7).

Several works mitigate popularity bias in the presence of implicit feedback [71, 70, 121, 123, 124, 125, 126, 127, 128, 129, 130, 131, 128]. Implicit feedback, such as clickstream data, purchase history, or time spent on an item, may not always clearly indicate user preferences. Users may click on items for various reasons, such as curiosity or price comparison, without being interested. Furthermore, implicit feedback can lead to positive unlabeled problems [27]. The positive unlabeled problem emphasizes that while visited or interacted items are considered positive examples, all other items may be uninteresting and should not be treated as negative examples but marked as unlabeled ones. On the other hand, explicit feedback is a more reliable [68] and accurate estimate of the user's interest [69]. One possible explanation behind more accurate estimation accounts for extensive research on methods such as Likert scales or questionnaires to capture feedback [69]. In addition, explicit feedback can capture absolute positive and negative feedback, unlike implicit feedback, which only provides positive and relative feedback. Thus, we

in this thesis explore a recommendation model to reduce popularity bias under explicit ratings. Some existing explicit feedback techniques handling popularity bias promote the diversification of non-popular items [122, 132]. A naive diversification may lead to poor accuracy of the overall recommender systems and can also result in poor accuracy on non-popular items [133, 134, 128, 135]. Another method to tackle popularity bias is using demographic bias [136]. In this post-processing technique, items are divided into advantageous and disadvantageous groups to have a fair representation of items belonging to disadvantaged groups in the overall ranking [137]. The ranking list is prepared after obtaining ratings for all user-item pairs, which forms a major overhead. Further, their idea revolves around having equal representation in the ranking list. In contrast, in our work (Chapter 7), we emphasize giving fair chances to both popular and non-popular items and not enforce strict equal representation. The amount of representation for items in our work is decided based on balancing losses on popular and non-popular items.

The work closest for solving popularity bias in matrix factorization under explicit feedback is to mitigate the bias by using Inverse Propensity Scores (IPS) [8]. The score helps in generating a pseudo missing completely at random dataset by weighting all the observed ratings. Although IPS loss is proven to be an unbiased estimator, these methods majorly suffer from two problems. First, the IPS estimator might become biased if the propensity estimation model is not appropriately stated. Second, IPS estimators suffer from high variance as the inverse of the propensities might be substantial. To overcome these challenges, Saito [27] proposed an asymmetric tri-training technique. It involves three rating predictors, two of which create a pseudo-rating dataset, and the third trains the model on these pseudo-ratings. The main limitation is that it becomes impossible to estimate the ratings of all items accurately as the dataset size reduces after applying the technique. Thus, there is a need for an effective strategy to tackle popularity bias in matrix factorization and will be of interest for this thesis.

A different line of work for balancing popularity bias in group or session recommendation is in [138, 139, 140]. On the other hand, we design a method to tackle popularity bias in individual recommendation systems.

The complete set of notations discussed in this chapter is summarized in Table 2.3. We will use these notations consistently throughout the thesis.

2.3 Conclusion

This chapter discusses the background work on clustering and recommender systems. The first half of the chapter introduces the notations and definitions that will help understand the contributions in the field of fair clustering. We surveyed results in offline fair clustering literature focusing on two fundamental levels of fairness: group and individual fairness. We also provided a categorization of fair clustering algorithms across multiple dimensions, such as implementation stage, solution approaches, and time complexity. Further, we discussed different fair clustering algorithms, surveyed their performance guarantees, and

Table 2.3: The table summarizes the notations discussed throughout the chapter.

Notation	**Description**
$X \subseteq \mathbb{R}^h$	Finite set of data points
k	Number of clusters
$x_i \in X$	Data point from X in offline setting
$\mathcal{C}$	k-clustering
$C = \{c_j\}_{j=1}^{k}$	Set of cluster centers
$\phi : X \to C$	Assignment function
$\{C_1, C_2, \ldots, C_k\}$	Set of k clusters of data X
$d : X \times X \to \mathbb{R}^+ \cup \{0\}$	Distance function
$\mathbb{I}(\cdot)$	Indicator function
$L_p(X, \phi)$	Objective cost (or clustering cost)
p	Norm value
β	Approximation factor of vanilla (unfair) clustering
X_ℓ	Data points belonging to protected group value ℓ
n_ℓ	Number of data points belonging to group value ℓ
$\rho : X \to [m]$	Protected group mapping function to one of m group values
$\boldsymbol{\tau}$	A vector of dimension ℓ
$r(x_i)$	Fair radius of data point x_i in individual fairness
α	Approximation parameter in individual fairness (α-FR)
$x_t \in X$	Data point arriving at time t in online clustering
$k_{\texttt{actual}}$	Actual number of centers opened in online setup when target is k (or $k_{\texttt{actual}}$)
Z	Number of clients in federated data clustering
$X^{(z)}$	Data points available at client z
$C^g = \{c_1^g, c_2^g, \ldots, c_k^g\}$	Set of global centers in federated data clustering
$C^{(z)} = \{c_1^{(z)}, c_2^{(z)}, \ldots, c_k^{(z)}\}$	Local (or best) set of centers on data $X^{(z)}$
$\texttt{H} \leq k$	Heterogeneity level
$L_p^{(z)}(C)$	Objective cost on federated client on set of centers C
$\phi_C^{(z)}$	Assignment function at client z using center set C
$\boldsymbol{\mu}^{(z)}(C^g)$	Mean objective cost per data point when using C^g as set of centers and data as $X^{(z)}$
$\boldsymbol{\mu}(C^g)$	Mean value of $\boldsymbol{\mu}^{(z)}(C^g)$ across all clients
$\boldsymbol{\sigma}(C^g)$	Cost deviation per data point
$\mathcal{U}$	Set of users in recommender system
$\mathcal{I}$	Set of items in recommender system
$R = \mathcal{U} \times \mathcal{I}$	Rating matrix where each entry $R_{u,i}$ corresponds to the true rating of item i by the user u on a scale of 1 (lowest) to 5 (highest).
P	Prediction matrix
δ	Loss function in recommender system
κ	Latent factor in matrix factorization
$\xi_u \in \mathbb{R}^\kappa$	User embedding vector for user u
$\psi_i \in \mathbb{R}^\kappa$	Item embedding vector for item i
η	Learning rate
$\mathcal{I}_\mathcal{P}$	Set of popular items
$\mathcal{I}_{\mathcal{NP}}$	Set of non-popular items

identified their limitations. Next, we provide an overview of existing works in online and federated setups, along with needed definitions and notations. These notations will be used consistently throughout the thesis. In the chapter's later part, we discussed recommender system preliminaries. We particularly provided an overview of popular matrix factorization algorithms and discussed the problem of popularity bias. We now provide **research gaps** below that will be the focus of the contributions in the thesis:

1. Different group fairness notions arose independently in literature. However, no existing study systematically examines the relationship between these notions.

2. There is no polynomial time algorithm for solving group fairness in offline clustering.

3. Many real-world applications demand the need to handle continuous incoming streams of data. In such scenarios, recomputing offline solutions can become computationally expensive and may even result in changing the data points' assignments in each execution. Thus, there is a need to handle data points online. Also, large-scale data may sometimes be distributed across different sites. Existing techniques in both online and distributed setups handle clustering, but no existing works handle fairness in online and federated settings.

4. Past literature shows that satisfying strict levels of multiple levels of fairness, say group and individual fairness, may not go hand in hand. Satisfying one level of fairness might result in lowering the other fairness level. A study that develops techniques to trace the Pareto frontier or help in achieving the user's desired level of fairness can be an interesting direction.

5. Plethora of literature has investigated popularity bias in implicit feedback. A few methods have come up to handle popularity bias in explicit feedback. However, devising methods that do not naively focus on increasing the diversity of non-popular items in the recommendation list is another interesting direction to improve current state-of-the-art approaches.

As part of this thesis, we will try to address these gaps in the next chapters.

Chapter 3

Group Fair Notion and Algorithms in Offline Clustering

Abstract

We revisit the problem of fair clustering in offline setting, first introduced by Chierichetti et al. [42], which requires each protected group to have approximately equal representation in every cluster, i.e., a Balance *property. Existing solutions to fair clustering are either not scalable or do not achieve an optimal trade-off between clustering objectives and fairness. In this chapter, we propose a new notion of fairness, which we call τ-ratio fairness, that enables a fine-grained efficiency vs. fairness trade-off. We also study the relationship between existing group fairness notions and τ-ratio fairness. We show that τ-ratio fairness is a stricter notion, and satisfying τ-ratio implies satisfying other existing notions. Furthermore, we show that a simple greedy round-robin-based algorithm achieves this trade-off efficiently. Under a more general setting of multi-valued protected groups, we rigorously analyze the theoretical properties of the proposed algorithm, Fair Round-robin Algorithm for Clustering Over End ($FRAC_{OE}$). We further propose a heuristic algorithm, Fair Round-robin Algorithm for Clustering (FRAC), that applies round-robin allocation at each iteration of the vanilla clustering algorithm. Our experimental results suggest that both FRAC and $FRAC_{OE}$ outperform all the state-of-the-art algorithms and work exceptionally well even for a large number of clusters.*

3.1 Introduction

The recent advancements in Machine Learning (ML) have led to the development of highly accurate models, leading to wide-scale adoption. ML models are being deployed in applications ranging from self-driving cars, approving home loan applications, criminal risk prediction, college admissions, and health risk prediction. The primary objective of these algorithms has been accuracy improvement. But their use to allocate social goods and opportunities such as access to healthcare, jobs, and education warrants a closer look at the societal impacts of their outcomes [143, 144]. Recent studies have

A preliminary part of this chapter has appeared in [141] (AAMAS 2023; as Extended Abstract) and [142] (GAIW Workshop Paper at AAMAS 2023). A detailed version of this chapter is published in DMKD Journal [44], and some parts are accepted as a book chapter in [43]. The work got appreciation as the Best Paper Award at the International Conference on Deployable AI 2022.

exposed a discriminatory outlook on the outcomes of these algorithms. The outcomes resulting from ML models are observed to have disparity in treatment towards individuals belonging to marginalized groups based on gender and race in real-world applications like automated resume processing [145], loan application screening, and criminal risk prediction [7]. Thus, designing fair and accurate machine learning models is an essential and immediate requirement for these algorithms to make a meaningful impact in the real world.

While fairness in supervised learning is well studied [146, 18, 147, 20, 136, 148], fairness in unsupervised learning is still in its formative stages [149, 150]. To emphasize the importance of fairness in unsupervised learning, we consider the following: An employee-friendly company is looking to open multiple branches across the city and distribute its workforce in these branches. The goal is to improve work efficiency and minimize overall travel time to work. The company has employees with varied backgrounds (race and gender) and does not prefer any group of employees over other groups. The company's diversity policy dictates hiring a minimum fraction of employees from each group in every branch. Thus, the natural question is: where should the branches be set up to maximize work efficiency, minimize travel time, and maintain diversity? In other words, the problem is to devise an unsupervised learning algorithm for identifying branch locations with the fairness (diversity) constraints applied to each branch. This problem can be naturally formulated as a clustering problem with additional fairness constraints on allocating the data points to the cluster centers (office locations).

Typically, fairness in supervised learning is measured by the algorithm's performance over different groups based on protected (sensitive) groups such as gender, race, and ethnicity. Motivated by this, the first fairness notion for clustering was proposed by Chierichetti et al. [42], wherein each cluster is required to exhibit a Balance, defined as the minimum ratio of protected and non-protected groups in any cluster. Their methodology, apart from having significant computational complexity, applies only to binary-valued protected groups. Further, it does not allow for trade-offs between the clustering objective and fairness guarantees. The subsequent literature ([11, 78, 151, 83]) improves efficiency; however, do not facilitate the explicit choice of the trade-off between the clustering objective cost and the fairness guarantee.

In this chapter, we define a new notion of fairness, which we call $\boldsymbol{\tau}$-ratio fairness. It ensures a certain fraction of data points for a given protected group in each cluster. We show that this simple notion of fairness has several advantages. First, the definition of $\boldsymbol{\tau}$-ratio naturally extends to multi-valued protected groups; second, $\boldsymbol{\tau}$-ratio fairness has closed-form theoretical relations to existing group fairness notions; third, it admits an intuitive and computationally efficient round-robin approach to fair allocation; fourth, it is straightforward for the algorithm designer to input the requirement into the algorithm as constraints; fifth, it is easy to interpret and evaluate it from the output. In our running example, if a company wants to have a minimum fraction of employees from each group in every branch (clusters), then one can simply specify it in the form of a vector $\boldsymbol{\tau}$ of size equal

to a number of protected groups. Through rigorous theoretical analysis, we show that the proposed algorithm FRAC$_{OE}$ provides a $2(\beta+2)$-approximate guarantee on the objective cost with $\boldsymbol{\tau}$-ratio fairness guarantee up to three clusters. Here, β is the approximation factor achieved by the vanilla clustering algorithm. We further experimentally demonstrate that our approach can achieve better clustering objective costs than any state-of-the-art (SOTA) approach on real-world data sets, even for a large number of clusters. Overall, the following are the contributions of our work. Overall, the following are the contributions of our work.

3.1.1 Our Contribution

Conceptual Contribution We introduce a new notion of fairness we call a $\boldsymbol{\tau}$-ratio fairness and show that any algorithm satisfying a $\boldsymbol{\tau}$-ratio fairness also satisfies the Balance property (Lemma 3.1). Also, we show that every parameter setting of Balance collapses to a degenerate value of $\boldsymbol{\tau}$-ratio fairness. The strictness of the proposed notion. We further propose two simple and efficient round-robin-based algorithms for the $\boldsymbol{\tau}$-ratio fair allocation problem, namely, FRAC$_{OE}$ (see, Section 3.4) and a heuristic algorithm called FRAC (Section 3.6). Our algorithms use the unconstrained clustering algorithm (referred to as vanilla clustering algorithm) as a black-box implementation and modify its output appropriately to ensure $\boldsymbol{\tau}$-ratio fairness. The fairness guarantee is deterministic and verifiable, i.e., holds for every run of the algorithm, and can be verified from the outcome without explicit knowledge of the underlying clustering algorithm. The guarantee on objective cost, however, depends on the approximation guarantee of the clustering algorithm. Our algorithms can handle multi-valued protected groups, allow user-specified bounds on Balance, are computationally efficient, and incur only an additional time complexity of $O(kn \log n)$, best in the current literature. Here, n is the total number of data points (dataset size), and k is the number of clusters.

Theoretical Contributions We show theoretical guarantees for our first algorithm; FRAC$_{OE}$. First, we show that FRAC$_{OE}$ achieves $2(\beta + 2)$-approximate for clustering instances up to three clusters (Theorem 3.11 and Lemma 3.15) with respect to optimal fair clustering cost for maximally balanced clusters; here β is a clustering algorithm specific constant. That is, given a fair clustering instance with $k \leq 3$ clusters and n data points, our proposed algorithm returns an allocation that has an objective cost of $2(\beta + 2)$ times the objective cost of optimal assignment with respect to optimally balanced clusters. We further show that this guarantee is tight (Proposition 3.16). For $k > 3$ clusters we show $2^{k-1}(\beta + 2)$-approximation guarantee on the $\boldsymbol{\tau}$-ratio. We conjecture that the exponential dependence of the approximation guarantee on k can be reduced to a constant. The proof for guarantees is extended to work for any general $\boldsymbol{\tau}$ vector (see Section 3.5.2). We also analyze the convergence of FRAC$_{OE}$ (Lemma 3.18) and provide the relationship between existing group fairness notions and $\boldsymbol{\tau}$-ratio fairness. To the best of our knowledge, we are first to show such relationships (Theorems 3.1 to 3.5).

Experimental Contributions Through extensive experiments on four datasets (Adult, Bank, Diabetes, and Census II), we show that the proposed algorithms, FRAC and FRAC$_{OE}$ outperform all the existing algorithms on fairness and objective costs. Perhaps the most important insight from our experiments is that the performance of our proposed algorithms does not deteriorate with increasing k. This experimentally validates our conjecture. Experiments also show that while we do not have convergence guarantees for heuristic algorithm FRAC, it does converge on all the datasets and performs slightly better than FRAC$_{OE}$. Thus making it suitable for practical applications. We compare our algorithms with `SOTA` algorithms for their fairness guarantee, objective cost, and runtime analysis. We also note that our algorithms do not require hyperparameter tuning, making our method easy to train and scalable. We demonstrate the efficacy of our algorithms using k-means and k-median. In addition to experimental validation of our proposed algorithms, we also validate our established theoretical relationships between different existing fairness notions. We show that satisfying $\boldsymbol{\tau}$-ratio fairness induces a certain level of existing group fairness notions.

3.2 Related Work

There is abundant literature on fairness in supervised learning [147, 152, 153, 154, 155, 156, 157]. But research on fair clustering is still in its infancy and is rapidly gathering attention [158, 49, 159, 160, 46, 161, 150, 162, 163]. These studies include extending the existing fairness levels such as group, individual fairness to clustering [12, 91, 54], proposing new problem-specific fairness levels such as social fairness [62, 48], characterizing the fairness versus efficiency trade-off [10, 93], developing and analyzing efficient fair algorithms [82, 78]. Among these, group fairness in clustering has been studied in various settings, including dynamic [164], capacitated [165], bounded cost [94], budgeted [166], privacy preserving [98], probabilistic [167], correlated [168], diversity aware [60], hierarchical, graph spectral, hypergraph [169, 170, 45], deep [171, 172, 173], distributed environments [174]. In this chapter, we focus on handling group fairness in offline clustering. The fairness in the offline setup has been introduced at different stages of implementation, namely – pre-processing, in-processing and post-processing are discussed separately below:

Pre-processing: Following a disparate impact doctrine [175], Chierichetti et al. [42], in their pioneering work, define fairness in clustering through a Balance property. Balance is defined as the ratio of data points with different protected group values in a cluster. A maximally balanced clustering ensures that the Balance in all the clusters is equal to the Balance in the original dataset (see Definition 3.2). Chierichetti et al. [42] achieves balanced clustering through the partitioning of the data into balanced sets called fairlets. It is followed by the merging of these partitions. Subsequently, Backurs et al. [11] propose an efficient algorithm to compute the fairlets. Both approaches have two major drawbacks: they are limited to the datasets having only binary-valued protected groups and can only create clusters exhibiting the exact Balance present in the original dataset (dataset ratio).

Thereby, they are not being flexible in achieving an optimal trade-off between Balance and accuracy. Chhabra et al. [176] recently devised the idea to use a pre-processing technique by the addition of a small number of extra data points called antidotes. Vanilla clustering techniques applied to this augmented dataset result in fair clusters with respect to the original data. The pre-processing technique to add antidotes requires solving a bi-level optimization problem. Furthermore, Schmidt et al. [78] extends the notion of coresets to fair clustering. They provide an efficient and scalable algorithm using *composable* fair coresets (see also [83, 151, 82, 177]). A coreset is a set of data points approximating the optimal clustering objective value for any k cluster centers. Though the coreset construction can be performed in a single pass over the data, storing them takes exponential space in terms of the dimension of the dataset. Bandyapadhyay et al. [82] though reduces this exponential size requirement to linear in terms of space, it still has a running complexity that is exponential in the number of clusters. Our proposed algorithms are efficient because we do not need any additional space. Simultaneously, the running complexity is linear in the number of clusters and near-linear in the number of data points.

In-processing: Böhm et al. [95] propose an (β+2)-approximate algorithm for fair clustering using a minimum cost-perfect matching algorithm. While the approach works with a multi-valued protected group, it has O(n^3) time complexity and is not scalable. Here, n is the number of data points in the dataset. Ziko et al. [10] propose a variational framework for fair clustering. Apart from being applicable to datasets with multi-valued protected group, the approach works for both prototype-based (k-mean/k-median) and graph-based clustering problems (N-cut or Ratio-cut [103]). However, the sensitivity of the hyper-parameter to various datasets and the number of clusters necessitates extensive tuning. This renders the approach computationally expensive. Further, the clustering objective also deteriorates significantly under strict fairness constraints when dealing with many clusters (k) (refer Section 3.7.1). Along the same lines, Abraham et al. [93] devise an optimization-based approach for fair clustering with multiple multi-valued protected groups. It has a trade-off hyper-parameter similar to [10].

Post-processing: Bera et al. [12] converted fair clustering into a fair assignment problem and formulated a linear programming (LP) based solution. The LP-based formulation leads to a higher execution time (refer to Section 3.7.4). Also, the approach fails to converge when dealing with a large number of clusters (k). The work by Bera et al. [12] is extended by Harb and Lam [13] for the 'k-center problem, whereas we consider k-means and k-median based centering techniques. Similarly the works in ([53, 178, 179, 180, 89, 63]) are applicable only for k-center clustering. Our proposed approach takes a similar route as Bera et al. [12] to convert the fair clustering problem into a fair allocation problem. However, we give a simple polynomial-time algorithm which, in $O(nk \log n)$ additional computations, guarantees $\boldsymbol{\tau}$-ratio fairness. Our allocation algorithms have the following main advantages over the current state of the art:

1. they are computationally efficient,

2. they work for multi-valued protected groups,

3. no hyperparameter tuning is required and,

4. they are simple and more interpretable (refer Section 3.3).

3.3 Preliminaries

Let $X \subseteq \mathbb{R}^h$ be a finite set of data points that need to be partitioned into k clusters. Each data point $x_i \in X$ is a feature vector described using h real-valued features. A k-clustering[1] $\mathcal{C} = (C, \phi)$ produces a partition of X into k subsets indexed by $[k] = \{1, 2, \ldots, k\}$. The clustering ($\mathcal{C}$) is characterized by a set of centers $C = \{c_j\}_{j=1}^{k}$, an assignment function $\phi : X \to C$ that maps each data point to the corresponding cluster center forming clusters $\{C_1, C_2, \ldots, C_k\}$ respectively. Furthermore, let $d : X \times X \to \mathbb{R}^+ \cup \{0\}$ denote a distance function obeying triangular inequality that measures the dissimilarity between features. Let $\mathbb{I}(\cdot)$ denote the indicator function, which takes a value of one if the condition inside the function is obeyed; otherwise, results in zero. Throughout this chapter, we consider that each point, $x_i \in X$ is associated with a *single* protected group $\rho(x_i)$ (say ethnicity from a pool of other available protected groups) that takes values from the set of m values denoted by $[m]$. The number of distinct protected group values is finite and much smaller than the size of the dataset[2]. Note that the protected group usually corresponds to disadvantaged groups and is known apriori to the algorithms as an additional input. Additionally, we are also given a vector $\boldsymbol{\tau} = \{\tau_\ell\}_{\ell=1}^{m}$, where each component τ_ℓ satisfies $0 \leq \tau_\ell \leq \frac{1}{k}$ and denotes the fraction of data points from the protected group value $\ell \in [m]$ required to be present in each cluster. An end-user can simply specify an m-dimensional vector with values between 0 and $1/k$ as the fairness target. Also, let us denote X_ℓ and n_ℓ as set and number of points, respectively, having protected group value ℓ in X. A vanilla (an unconstrained) clustering algorithm determines the cluster centers to minimize the following objective cost:

Definition 3.1 (Objective Cost)

Given p, the clustering objective cost with respect to the metric space (X, d) is defined as:

$$L_p(X, \phi) = \left(\sum_{x_i \in X} d(x_i, \phi(x_i))^p \right)^{\frac{1}{p}} \tag{3.1}$$

Different values of p, will result in different objective cost: $p = 1$ for k-medians, $p = 2$ for k-means, and $p = \infty$ for k-centers. Our aim is to develop an algorithm that minimizes the objective cost irrespective of p while ensuring fairness.

Group Fairness Notions: We begin with re-defining the most popular notion of group fairness called Balance.

[1] Throughout the thesis, for simplicity, we call a k-clustering simply clustering.

[2] Otherwise, the problem is uninteresting as balanced clustering may not be feasible.

Definition 3.2 (τ-Balance)

For a binary valued protected group taking values from set $\{\ell_1, \ell_2\}$, a clustering $\mathcal{C}$ is said to be τ-Balance [42] with

$$\tau = \min_{C_j \in \mathcal{C}} \left(\min \left(\frac{\sum_{x_i \in C_j} \mathbb{I}(\rho(x_i) = \ell_1)}{\sum_{x_i \in C_j} \mathbb{I}(\rho(x_i) = \ell_2)}, \frac{\sum_{x_i \in C_j} \mathbb{I}(\rho(x_i) = \ell_2)}{\sum_{x_i \in C_j} \mathbb{I}(\rho(x_i) = \ell_1)} \right) \right). \tag{3.2}$$

A generalization of Balance to multi-valued protected groups is proposed by Bera et al. [12] in terms of cluster sizes.

Definition 3.3 ($\boldsymbol{\tau}$-MP)

A clustering $\mathcal{C}$ is said to obey **minority protection** with respect to $\boldsymbol{\tau}$ (i.e. $\boldsymbol{\tau}$-MP [12]) if for all $\ell \in [m], C_j \in \mathcal{C}$,

$$\sum_{x_i \in C_j} \mathbb{I}(\rho(x_i) = \ell) \geq \tau_\ell |C_j|. \tag{3.3}$$

The minority protection constraints the lower bound on the number of data points from each protected group in every cluster.

Definition 3.4 ($\boldsymbol{\tau}$-RD)

A clustering $\mathcal{C}$ is said to obey **restricted dominance** with respect to $\boldsymbol{\tau}$ (i.e. $\boldsymbol{\tau}$-RD [12]) if for all $\ell \in [m], C_j \in \mathcal{C}$,

$$\sum_{x_i \in C_j} \mathbb{I}(\rho(x_i) = \ell) \leq \tau_\ell |C_j|. \tag{3.4}$$

Restricted dominance constraints the upper on the number of data points from each protected group in every cluster.

For binary protected group taking values $a, b \in [m]$ with $\tau_a = \tau_b = \min_{a,b} \frac{n_a}{n_b}$, this notion becomes exactly same as the τ-Balance notion. Hence, minority protection along with restricted dominance generalizes Balance notion to a multi-valued protected group. We now define our proposed $\boldsymbol{\tau}$-ratio fairness notion, which ensures that each cluster has a predefined fraction of data points for each protected group value. $\boldsymbol{\tau}$-ratio requires only priorly known dataset composition, which helps achieve polynomial-time algorithms.

Definition 3.5 ($\boldsymbol{\tau}$-ratio Fairness)

An assignment function ϕ satisfies $\boldsymbol{\tau}$-ratio fairness if

$$\sum_{x_i \in C_j} \mathbb{I}(\rho(x_i) = \ell) \geq \tau_\ell \sum_{x_i \in X} \mathbb{I}(\rho(x_i) = \ell) \ \forall C_j \in \mathcal{C} \text{ and } \forall \ell \ \in [m] \tag{3.5}$$

The notion of τ-Balance, τ-MP, and τ-RD defines the fairness with respect to the data points within a cluster (unknown a-priori) corresponding to different group values. The

$\boldsymbol{\tau}$-FAIR notion on other hand imposes restrictions on the number of data points from each group within a cluster to the overall data points of the respective group in the dataset. our fairness notion ($\boldsymbol{\tau}$-ratio) resembles that of balanced (in terms of number of data points in each cluster) clustering studied by Banerjee and Ghosh [181] without fairness constraint. However, their proposed sampling technique is not designed to guarantee $\boldsymbol{\tau}$-ratio fairness and does not analyze loss incurred due to having these fairness constraints. We now discuss the relationship between τ-BALANCE, τ-MP, and τ-RD to $\boldsymbol{\tau}$-ratio fairness.

3.3.1 Relationship between Group Fairness Notions

While the different fairness notions arose independently in the literature, we show that they are related when dealing with a binary protected group (i.e., takes only two values $a, b \in [m]$) as illustrated in Figure 3.1. Our first lemma shows that an algorithm satisfying $\boldsymbol{\tau}$-ratio fairness produces a set of clusters that also achieves a certain Balance. In particular, when $\tau_\ell = \frac{1}{k}$, then $\boldsymbol{\tau}$-ratio fairness achieve the Balance equal to the *dataset ratio*.

Lemma 3.1. *If a cluster $C_j \in \mathcal{C}$ is $\boldsymbol{\tau}$-ratio fair, then it also satisfies* $\min_{a,b}\left(\frac{\tau_a}{1-k\tau_b+\tau_b}\frac{n_a}{n_b}\right)$-BALANCE *where n_a, n_b are the total number of data points for group a, b respectively. Further when τ_a=τ_b=$1/k$ in $\boldsymbol{\tau}$-ratio fairness then it is* $\min_{a,b}(n_a/n_b)$-Balance *clustering.*

Proof. Given n_a, suppose an algorithm satisfies $\boldsymbol{\tau}$-ratio fairness then for any cluster C_j and protected group value a, we have:

$$\tau_a n_a \leq \sum_{x_i \in C_j} \mathbb{I}(\rho(x_i) = a) \leq n_a(1 - k\tau_a + \tau_a) \tag{3.6}$$

Here, the lower bound comes directly from the fairness definition and the upper bound is derived from the fact that all the clusters together will be allocated at least $k\tau_a n_a$ number of data points. The extra data points that a particular cluster can take are upper bounded by $n_a - kn_a\tau_a$. Thus, the τ-BALANCE of the cluster with respect to the two values a and b should follow

$$\frac{\sum_{x_i \in C_j} \mathbb{I}(\rho(x_i) = a)}{\sum_{x_i \in C_j} \mathbb{I}(\rho(x_i) = b)} \geq \frac{\tau_a n_a}{n_b(1 - k\tau_b + \tau_b)} \tag{3.7}$$

□

Lemma 3.1 shows that one can achieve the desired amount of τ'-BALANCE by appropriately setting $\boldsymbol{\tau} = \{\tau_a, \tau_b\}$. When $\tau_a = \tau_b = 1/k$, then we get $\min_{a,b}(n_a/n_b)$-Balanceand give the following corollary:

Corollary 3.2. *For $\tau_a = \tau_b = \frac{1}{k}$, $\boldsymbol{\tau}$-ratio fairness guarantee ensures the* dataset ratio *for all the clusters.*

We now show that the converse is not true. That is, a clustering satisfying Balance (equal to dataset ratio) can result in arbitrary bad $\boldsymbol{\tau}$-ratio fairness.

Lemma 3.3. *A fair clustering instance exists which satisfies τ'-*BALANCE *with $\tau' > 0$ and has arbitrarily low $\boldsymbol{\tau}$-ratio.*

Suppose in a $k(=2)$-clustering instance the binary protected group takes values a, b such that $n_a{=}n_b{=}n/2$. Now, let one data point from each group be allocated to cluster 1 and the remaining data points to cluster 2. Then, we have $\tau_a = \tau_b{=}\ 2/n$ which can go arbitrarily small for large n.

From Lemma 3.1, 3.3 we see that $\boldsymbol{\tau}$-ratio is more stricter than τ-BALANCE. Also, both notions behave conceptually differently in how they induce fair clusters. Since the Balance does not add any constraint on cluster size and requires only a minimum representation ratio, which might result in skewed clusters. However, $\boldsymbol{\tau}$-ratio leads to a controlled distribution of data points among clusters.

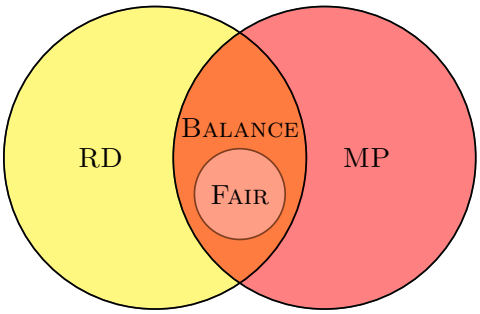


Figure 3.1: Relationship between the different group fairness notions.

Lemma 3.4. *The cluster satisfying both $\boldsymbol{\tau}'$-MP and $\boldsymbol{\tau}$-RD ensures* $\min\left(\frac{\tau'_a}{\tau_b}, \frac{\tau'_b}{\tau_a}\right)$*-*BALANCE. *Furthermore, satisfying only one of them does not ensure τ-*BALANCE.

Proof. From the definition of $\boldsymbol{\tau}$-RD and $\boldsymbol{\tau}'$-MP, $\forall C_j \in \mathcal{C}$ we get

$$\tau'_a \leq \frac{\sum_{x_i \in C_j} \mathbb{I}(\rho(x_i) = a)}{|C_j|} \leq \tau_a \text{ and } \tau'_b \leq \frac{\sum_{x_i \in C_j} \mathbb{I}(\rho(x_i) = b)}{|C_j|} \leq \tau_b \tag{3.8}$$

$$\text{So } \frac{\tau'_a}{\tau_b} \leq \frac{\sum_{x_i \in C_j} \mathbb{I}(\rho(x_i) = a)}{\sum_{x_i \in C_j} \mathbb{I}(\rho(x_i) = b)} \leq \frac{\tau_a}{\tau'_b} \text{ and } \frac{\tau'_b}{\tau_a} \leq \frac{\sum_{x_i \in C_j} \mathbb{I}(\rho(x_i) = b)}{\sum_{x_i \in C_j} \mathbb{I}(\rho(x_i) = a)} \leq \frac{\tau_b}{\tau'_a} \tag{3.9}$$

Thus, from the above equations we can say, τ-BALANCE $\geq min\left(\frac{\tau'_a}{\tau_b}, \frac{\tau'_b}{\tau_a}\right)$

□

Lemma 3.5. *If a cluster satisfies τ-*BALANCE *then it is also $\boldsymbol{\tau}$-MP with $\boldsymbol{\tau}{=}\{\frac{1}{2}, \frac{\tau}{1+\tau}\}$ and $\boldsymbol{\tau}$-RD with $\boldsymbol{\tau}{=}\{\frac{1}{1+\tau}, \frac{1}{2}\}$ for $\{a, b\}$ respectively.*

Proof. Given τ-BALANCE $= min\left(\frac{\tau_a}{\tau_b}, \frac{\tau_b}{\tau_a}\right)$. Without loss of generality let us assume that $\tau = \frac{\tau_a}{\tau_b}$ that is $\tau_a{=}\ \tau\ (\tau_b)$. Since τ is the minimum value over all clusters, for any arbitrary cluster containing τ'_a and τ'_b data points, we have $\frac{\tau_a}{\tau_b} \leq \frac{\tau'_a}{\tau'_b}$.

Since the upper-bound of τ can be 1 (perfectly balanced clusters), we have $\tau \leq \frac{\tau'_a}{\tau'_b} \leq 1$. Adding 1 on both sides,

$$\tau + 1 \leq \frac{\tau'_a + \tau'_b}{\tau'_b} \leq 2 \implies \frac{1}{2} \leq \frac{\tau'_b}{\tau'_a + \tau'_b} \leq \frac{1}{1+\tau} \tag{3.10}$$

So, the $\boldsymbol{\tau}$-MP and $\boldsymbol{\tau}$-RD for b is $\frac{1}{2}$, $\frac{1}{1+\tau}$ respectively. Similarly,

$$1 \leq \frac{\tau'_b}{\tau'_a} \leq \frac{1}{\tau} \implies 2 \leq \frac{\tau'_b + \tau'_a}{\tau'_a} \leq \frac{1}{\tau} + 1 \implies \frac{\tau}{1+\tau} \leq \frac{\tau'_a}{\tau'_b + \tau' a} \leq \frac{1}{2} \tag{3.11}$$

So, the $\boldsymbol{\tau}$-MP and $\boldsymbol{\tau}$-RD for a is $\frac{\tau}{1+\tau}$, $\frac{1}{2}$ respectively.
Thus, in all we have $\boldsymbol{\tau}$-MP with $\boldsymbol{\tau}$=$\{\frac{1}{2}, \frac{\tau}{1+\tau}\}$ and $\boldsymbol{\tau}$-RD with $\boldsymbol{\tau}$=$\{\frac{1}{1+\tau}, \frac{1}{2}\}$ for $\{a, b\}$ respectively. □

Lemma 3.1 and Lemma 3.5 lead us to following corollary.

Corollary 3.6. *If a cluster satisfies $\boldsymbol{\tau}$-ratio then it is also $\boldsymbol{\tau}'$-MP with $\boldsymbol{\tau}'$=$\{\frac{1}{2}, \frac{\tau_a n_a}{(1-k\tau_b+\tau_b)n_b+\tau_a n_a}\}$ and $\boldsymbol{\tau}'$-RD with $\boldsymbol{\tau}'$=$\{\frac{(1-k\tau_b+\tau_b)n_b}{(1-k\tau_b+\tau_b)n_b+\tau_a n_a}, \frac{1}{2}\}$ where Lemma 3.1 is say minimum over group value $a \in [m]$.*

The example for Lemma 3.3, also satisfies $\boldsymbol{\tau}$-MP and $\boldsymbol{\tau}$-RD with $\boldsymbol{\tau}$=$\{1\}$ and $\{n/2-1\}$ respectively. However, $\boldsymbol{\tau}$-ratio can again go arbitrarily low. So we get the corollary as follows.

Corollary 3.7. *A fair clustering can exist that obeys $\boldsymbol{\tau}$-MP and $\boldsymbol{\tau}$-RD but can have arbitrarily low $\boldsymbol{\tau}$-ratio fairness.*

All the above results prove that $\boldsymbol{\tau}$-ratio is a more stricter notion. Thus, we focus on designing an algorithm satisfying $\boldsymbol{\tau}$-ratio fairness while minimizing objective cost irrespective of p. To this, we now define the fair clustering problem with respect to the proposed fairness notion:

Definition 3.6 ($\boldsymbol{\tau}$-ratio Fair Clustering Problem)

The objective of a $\boldsymbol{\tau}$-ratio fair clustering problem $\mathcal{I}$ is to estimate $\mathcal{C} = (C, \phi)$ that minimizes the objective cost $L_p(X, \phi)$ subject to the $\boldsymbol{\tau}$-ratio fairness guarantee. The optimal objective cost of a $\boldsymbol{\tau}$-ratio fair clustering problem is denoted by $\mathcal{OPT}_{clust}(\mathcal{I})$.

A solution to this problem is to rearrange the data points (learn a new ϕ) with respect to the cluster centers obtained after a traditional clustering algorithm (called vanilla clustering) to guarantee $\boldsymbol{\tau}$-ratio fairness. The problem of rearrangement of data points with respect to the fixed centers is known as the fair assignment problem, which we define below:

Definition 3.7 ($\boldsymbol{\tau}$-ratio Fair Assignment Problem)

Given X and $C = \{c_j\}_{j=1}^{k}$, the solution to the fair assignment problem $\mathcal{T}$ produces an assignment $\phi : X \to C$ that ensures $\boldsymbol{\tau}$-ratio fairness and minimizes $L_p(X, \phi)$. The optimal objective function value to a $\boldsymbol{\tau}$-ratio fair assignment problem is denoted by $\mathcal{OPT}_{assign}(\mathcal{T})$.

However, this transformation of the fair clustering problem $\mathcal{I}$ into a fair assignment problem $\mathcal{T}$ should ensure that $\mathcal{OPT}_{assign}(\mathcal{T})$ is not too far from $\mathcal{OPT}_{clust}(\mathcal{I})$. The connection between fair clustering and fair assignment problem is established through the following lemma.

Lemma 3.8. *Let $\mathcal{I}$ be an instance of a fair clustering problem and $\mathcal{T}$ an instance of $\boldsymbol{\tau}$-ratio fair assignment problem after applying an β-approximate solution to the vanilla clustering problem, then $\mathcal{OPT}_{assign}(\mathcal{T}) \leq (\beta + 2)\mathcal{OPT}_{clust}(\mathcal{I})$.*

Proof. Let C be the cluster centers obtained by running a vanilla clustering algorithm on instance $\mathcal{I}$. The proof of the Lemma depends on the existence of an assignment $\hat{\phi}$ satisfying $\boldsymbol{\tau}$-ratio fairness such that $L_p(X, \hat{\phi}) \leq (\beta + 2)\mathcal{OPT}_{clust}(\mathcal{I})$. Then it follows as $\mathcal{OPT}_{assign}(\mathcal{T}) \leq L_p(X, \hat{\phi}) \leq (\beta + 2)\mathcal{OPT}_{clust}(\mathcal{I})$.
To this, let (C^*, ϕ^*) denote the optimal solution to $\mathcal{I}$. Define $\hat{\phi}$ as follows: for every $c^* \in C^*$, let $nrst(c^*) = \text{argmin}_{c \in C} d(c, c^*)$ be the nearest center to c^*. Then, for every $x_i \in X$, define $\hat{\phi}(x_i) = nrst(\phi^*(x_i))$. Then we have the following two claims:

Claim 3.9. *$\hat{\phi}$ satisfies $\boldsymbol{\tau}$-ratio fairness.*

Proof. Let the set of data points having protected group value ℓ in cluster $c^* \in C^*$ be $n_\ell(c^*)$. Since (C^*, ϕ^*) satisfy $\boldsymbol{\tau}$-ratio fairness then using Definition 3.5 we have

$$|n_\ell(c^*)| \geq \tau_\ell n_\ell \quad \forall c^* \in C^*. \tag{3.12}$$

Now, for any center $c \in C$ belonging to vanilla clustering, we will find the set of all centers in optimal solution (C^*) that are nearest to c. Let us denote this set by $N(c) = \{c^* \in C^* : nrst(c^*) = c\}$.
Then the way $\hat{\phi}$ is defined, we have, $\forall c$:

$$|\{x_i \in X_\ell : \hat{\phi}(x_i) = c\}| = |\cup_{c^* \in N(c)} n_\ell(c^*)| \tag{3.13}$$

Now as each center c^* satisfies $\boldsymbol{\tau}$-ratio fairness, the union over combined assignments for each center in $N(c)$ and since each set of assignments satisfies $\boldsymbol{\tau}$-ratio so union will also satisfy $\boldsymbol{\tau}$-ratio fairness i.e. $|\cup_{c^* \in N(c)} n_\ell(c^*)| \geq n_\ell \tau_\ell$. □

Claim 3.10. $L_p(X, \hat{\phi}) \leq (\beta + 2)\mathcal{OPT}_{clust}(\mathcal{I})$.

Proof. The proof of this claim uses triangle inequality and is exactly the same as Claim 6 of Bera et al. [12]. We re-write it here using our set of notations for the sake of completeness.

Algorithm 1: τ-FRAC$_{OE}$

Input: set of datapoints X, number of clusters k, fairness requirement vector $\boldsymbol{\tau}$, range of protected group values m, clustering objective norm p

Output: cluster centers $\hat{C}$ and assignment function $\hat{\phi}$

```
Solve the vanilla (k, p)-clustering problem and let (C, φ) be the solution obtained.
if τ-ratio fairness is met then
    return (C, φ)
    else
        (Ĉ, φ̂) = FairAssignment(C, X, k, τ, m, p, φ)
        return (Ĉ, φ̂)
    end
end
```

Fix a data point $x_i \in X$. Let $c = \phi(x_i)$, $\hat{c} = \hat{\phi}(x_i)$, and $c^* = \phi^*(x_i)$. Then we have,

$$d(x_i, \hat{c}) = d(x_i, nrst(c^*)) \leq d(x_i, c^*) + d(c^*, nrst(c^*)) \leq d(x_i, c^*) + d(c^*, c) \leq 2d(x_i, c^*) + d(x_i, c) \tag{3.14}$$

The first and third step follows using the triangular inequality. While the second step is based on the definition of nrst. So, if we define assignment cost vectors corresponding to ϕ, $\hat{\phi}$ and ϕ^* as $\vec{d} = \{d(x_i, \phi) : x_i \in X\}$, $\vec{d'} = \{d(x_i, \hat{\phi}) : x_i \in X\}$ and $\vec{d^*} = \{d(x_i, \phi^*) : x_i \in X\}$ respectively. Then using the above bound, we get $\vec{d'} \leq 2\vec{d} + \vec{d^*}$. Now, since L_p is a monotone norm on these vectors,

$$L_p(X, \hat{\phi}) = L_p(\vec{d'}) \leq 2L_p(\vec{d}) + L_p(\vec{d^*}) = 2L_p(X, \phi^*) + L_p(X, \phi). \tag{3.15}$$

This completes the proof by using the fact that $L_p(X, \phi^*) = \mathcal{OPT}_{clust}(\mathcal{I})$ and $L_p(X, \phi) \leq \beta\mathcal{OPT}_{clust}(\mathcal{I})$. □

Both these claims complete the proof of the Lemma 3.8. □

A similar technique of converting fair clustering to a fair assignment problem was proposed by Bera et al. [12]. However, Bera et al. [12] proposed a linear programming-based solution to obtain the Balance fair assignment. Although the solution is theoretically strong, there are two issues with the algorithm. Firstly, the time complexity is high (as can be seen from the experiments in Section 3.7.4) and secondly, the solution obtained is not easy to interpret due to the use of the complicated linear program. By interpretability, we try to find the answer to the following question – Why is a data point assigned to a specific cluster to maintain fairness? We propose a simple round-robin (easily interpretable) FRAC$_{OE}$ algorithm for a fair assignment problem with a time complexity of $O(kn \log n)$ in the next section.

Algorithm 2: FairAssignment

Input: cluster centers C, set of datapoints X, number of clusters k, fairness requirement vector $\boldsymbol{\tau}$, range of protected group m, clustering objective norm p, assignment function ϕ

Output: Cluster centers $\hat{C}$ and assignment function $\hat{\phi}$

Fix a random ordering on centers and let the centers are numbered from 1 to k with respect to this random ordering.
Initialize $\hat{\phi}(x_i) \leftarrow 0 \ \forall x_i \in X$
for $\ell \leftarrow 1$ **to** m **do**
    $n_\ell \leftarrow$ number of data points having value of protected group ℓ.
    $X_\ell \leftarrow$ set of data points having value of protected group ℓ.
    for $t \leftarrow 1$ **to** $\tau_\ell n_\ell$ **do**
        for $j \leftarrow 1$ **to** k **do**
            $x_{min} \leftarrow \text{argmin}_{x_i \in X_\ell : \hat{\phi}(x_i) = 0} \, d(x_i, c_j)$
            $\hat{\phi}(x_{min}) = j$
        end
    end
    For all $x_i \in X_\ell$ such that $\hat{\phi}(x_i) = 0$, set $\hat{\phi}(x_i) = \phi(x_i)$
end
Recompute the centers $\hat{C}$ with respect to the new allocation function $\hat{\phi}$.
Return $(\hat{C}, \hat{\phi})$

3.4 Fair Round-robin Algorithm for Clustering Over End (FRAC_{OE})

Fair Round-robin Algorithm for Clustering Over End (FRAC_{OE}) first runs a vanilla clustering algorithm to produce the initial clusters $\mathcal{C} = (C, \phi)$. It then *makes corrections* as follows. The algorithm first checks if $\boldsymbol{\tau}$-ratio fairness is met with the current allocation ϕ, in which case it returns $\hat{\phi} = \phi$ and $\hat{C} = C$. If the assignment ϕ violates the $\boldsymbol{\tau}$-ratio fairness constraint then the new assignment function $\hat{\phi}$ is computed according to FairAssignment procedure in Algorithm 2.

Algorithm 2 iteratively allocates the data points with respect to each protected group value. To recollect X_ℓ and n_ℓ denote the set and the number of data points having ℓ as the protected group value, respectively. The algorithm allocates $\lfloor \tau_\ell n_\ell \rfloor$ number of data points [3] to each cluster in a round-robin fashion as follows. Let $\{c_1, c_2, \ldots, c_k\}$ be a random ordering of the cluster centers. At each round t, each center c_j picks the data point x_i of its preferred choice from X_ℓ i.e. $\hat{\phi}(x_i) = j$. Once the τ_ℓ fraction of data points are assigned to the centers, i.e., after $\tau_\ell n_\ell$ number of rounds, the allocation of remaining data points is set to its original assignment ϕ. Note that this algorithm will certainly satisfy $\boldsymbol{\tau}$-ratio fairness as, in the end, the algorithm assures that at least τ_ℓ fraction of data points are allotted to each cluster for a protected group value ℓ. We defer to theoretical results to assert the quality of the clusters. The runtime complexity of Algorithm 2 is $O(kn \log n)$

[3] For the sake of simplicity, we assume $\tau_\ell n_\ell \in \mathbb{N}$ and ignore the floor notation.

as step 4 requires the data points to be sorted in the increasing order of their distances with the cluster centers.

3.5 Theoretical Results

We now provide the theoretical guarantees of FRAC_{OE} with respect to $\boldsymbol{\tau}$-ratio fairness. We begin by providing guarantees for maximally balanced clusters, i.e. $\tau_\ell = 1/k \; \forall \ell \in [m]$.

3.5.1 Guarantees for FRAC_{OE} for $\tau = \{1/k\}_{l=1}^{m}$

Theorem 3.11. *Let $k = 2$ and $\tau_\ell = \frac{1}{k}$ for all $\ell \in [m]$. An allocation returned by $FRAC_{OE}$ guarantees $\boldsymbol{\tau}$-ratio fairness and satisfies 2-approximation guarantee with respect to an optimal fair assignment up to an instance-dependent additive constant.*

Proof. **Correctness and Fairness:** Clear from the construction of the algorithm.
Proof of (approximate) Optimality: We will prove 2-approximation with respect to each value ℓ of protected group separately.
We now show that $\text{FRAC}_{OE}(\mathcal{T}) \leq 2 \; \mathcal{OPT}_{assign}(\mathcal{T}) + \vartheta$, where $\text{FRAC}_{OE}(\mathcal{T})$ and $\mathcal{OPT}_{assign}(\mathcal{T})$ denote the objective value of the solution returned by FRAC_{OE} and optimal assignment algorithm respectively on given instance $\mathcal{T} = (C, X)$. Let $\vartheta := 2 \sup_{x,y \in X} d(x, y)$ be the diameter of the feature space. We begin with the following useful definition.

Definition 3.8 (Bad Assignments)

Let $\mathcal{C}_1$ and $\mathcal{C}_2$ represent the set of data points assigned to c_1 and c_2 by optimal assignment algorithm[a]. The i^{th} round (i.e. assignments g_i to c_1 and h_i to c_2) of FRAC_{OE} is called

- 1-bad if exactly one of 1) $g_i \notin \mathcal{C}_1$ or 2) $h_i \notin \mathcal{C}_2$ is true, and
- 2-bad if both 1) and 2) above are true.

Furthermore, a round is called bad if it is either 1-bad or 2-bad and called good otherwise.

[a]Note that an optimal fair allocation need not be unique. Our result holds for any optimal fair allocation.

Let all incorrectly assigned data points in a bad round be called bad assignments. We use the following convention to distinguish between different bad assignments. If $g_i \notin \mathcal{C}_1$ holds, we refer to it as type 1 bad assignment, i.e. if data point g_i is currently assigned to $\mathcal{C}_1$ but should belong to optimal clustering $\mathcal{C}_2$. Similarly, if $h_i \notin \mathcal{C}_2$ holds, it is a type 2 bad assignment, i.e. h_i should belong to optimal clustering $\mathcal{C}_1$ but is currently assigned to c_2. Hence a 2-bad round results in 2 bad assignments one of each type i.e. $g_i \notin \mathcal{C}_1$ and $h_i \notin \mathcal{C}_2$. In summary, each 1-bad round can have either type 1 or type 2 bad assignment,

and each 2−bad round will have two bad assignments each of type 1 and type 2. Finally, let B be the set of all bad rounds and A be the set of all bad assignments.

Definition 3.9 (Complementary Bad Pair)

A pair of data points $w, z \in A$ such that w is a bad assignment of type t and z is a bad assignment of type $|3-t|$ is called a complimentary bad pair if,
1) w and z are allocated in same round (i.e. in a 2-bad round) or
2) if they are allocated in i^{th} and j^{th} 1-bad rounds respectively with $i < j$, then z is the first bad assignment of type $(3-t)$ which has not been yet paired with a complementary assignment.

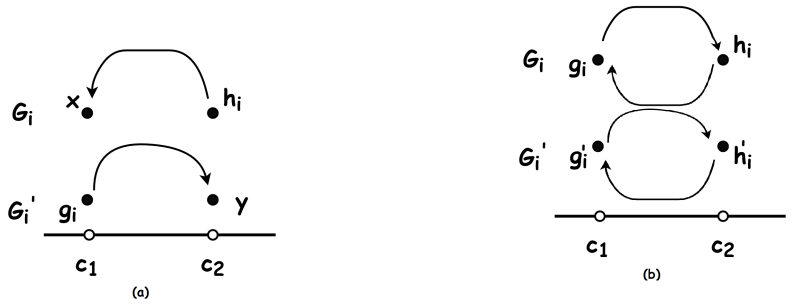
Lemma 3.12. *If n_ℓ is even, every bad assignment in the allocation returned by FRAC$_{OE}$ has a complementary assignment. If n_ℓ is odd, at most, one bad assignment will be left without a complementary assignment.*

Proof. Let $B = B_1 \cup B_2$, where B_t is a set of t-bad rounds. Note that the claim is trivially true if $B_1 = \emptyset$. Hence, let $|B_1| > 0$ and write $B_1 = B_{1,1} \cup B_{1,2}$. Here $B_{1,t}$ is a 1-bad round that resulted in type t bad assignment. Let $H_{1,t}$ be the set of good assignments of type t (i.e. correctly assigned to the center c_t) allocated in 1-bad rounds.
When n_ℓ is even, $|\mathcal{C}_1| = |\mathcal{C}_2|$ we have $|B_{1,2}| + |H_{1,1}| = |B_{1,1}| + |H_{1,2}|$. This is true because one can ignore good rounds and 2-bad rounds as every 2-bad round can be converted into a good round by switching the assignments. Further observe that, as FRAC$_{OE}$ assigns two data points per round and each round results in exactly one bad assignment and exactly one good assignment, we have $|H_{1,t}| = |B_{1,(3-t)}|$. Together, we have $|B_{1,1}| = \frac{|B_{1,2}|+|H_{1,1}|}{2} = |B_{1,2}|$. When n_ℓ is odd, we might have one additional data point left in the last 1-bad round that is not being assigned any complementary data point. This completes the proof of the lemma. □

We will bound the optimality of 1-bad rounds and 2-bad rounds separately.

Bounding 1-bad rounds: When n_ℓ is even, from Lemma 3.12, there are even numbers of 1-bad rounds; two for each complimentary bad pair. Let the 4 data points of corresponding two 1-bad rounds be $G_i : (x, h_i)$ and $G_i' : (g_i, y)$ as shown in Figure 3.2a. Note that $x \in \mathcal{C}_1$ and $y \in \mathcal{C}_2$ i.e. both are good assignments and $g_i \notin \mathcal{C}_1$, $h_i \notin \mathcal{C}_2$ are bad assignments. Now, consider an instance $\mathcal{T}_i = \{C, \{x, h_i, g_i, y\}\}$, then $\mathcal{OPT}_{assign}(\mathcal{T}_i) = d(x, c_1) + d(h_i, c_1) + d(g_i, c_2) + d(y, c_2)$. We consider, without loss of generality, that the round G_i takes place before G_i' in the execution of FRAC$_{OE}$. The proof is similar to the other case. First note that since FRAC$_{OE}$ assigns h_i to cluster 2 while both g_i and y were available, we have

$$d(h_i, c_2) \leq d(g_i, c_2) \text{ and } d(h_i, c_2) \leq d(y, c_2) \tag{3.16}$$



(a) Two 1-bad round pairs (b) Two 2-bad round pairs

Figure 3.2: Different cases for $k = 2$. (a) Shows two 1-bad rounds with four assignments such that x, y are good assignments and allocated to the optimal center by algorithm, whereas g_i and h_i are bad assignments with an arrow showing the direction to the optimal center from the assigned center. (b) Shows four bad data points such that g_i, g_i' are assigned to c_1 but should belong to c_2 in optimal clustering (the arrow depicts the direction to optimal center). Similarly, h_i, h_i' should belong to c_1 in optimal clustering.

So,

$$\begin{aligned}
&\text{FRAC}_{OE}(\mathcal{T}_i) && (3.17)\\
&= d(x,c_1) + d(h_i,c_2) + d(g_i,c_1) + d(y,c_2) && (3.18)\\
&\leq d(x,c_1) + d(h_i,c_2) + d(g_i,c_2) + d(c_1,c_2) + d(y,c_2) && (\because \text{triangle inequality})\\
&\leq d(x,c_1) + d(h_i,c_2) + d(g_i,c_2) + d(h_i,c_2) + d(h_i,c_1) + d(y,c_2) && (3.19)\\
&\leq d(x,c_1) + d(y,c_2) + d(g_i,c_2) + d(g_i,c_2) + d(h_i,c_1) + d(y,c_2) && (\because \text{Equation. 3.16})\\
&\leq 2\ \mathcal{OPT}_{assign}(\mathcal{T}_i) && (3.20)
\end{aligned}$$

If n_ℓ is odd, then all the other rounds can be bounded using the above cases except one extra 1-bad round. Let the two data points corresponding to this round G_i be (g_i, y). Thus, $\text{FRAC}_{OE}(\mathcal{T}_i) \leq 2\mathcal{OPT}_{assign}(\mathcal{T}_i) + \vartheta$. Here $\vartheta{=}2\sup_{x,y\in\mathcal{X}} d(x,y)$ is the diameter of the feature space.

Bounding 2-bad rounds: First, assume that there is an even number of 2-bad rounds. In this case consider the pairs of consecutive 2-bad rounds as $G_i : (g_i, h_i)$ and $G_i^{'} = (g_i', h_i')$ with G_i' bad round followed by G_i (Figure 3.2b). Note that $g_i, g_i' \in \mathcal{C}_2$ and $h_i, h_i' \in \mathcal{C}_1$. Now consider instance $\mathcal{T}_i = \{C, \{g_i, g_i', h_i, h_i'\}\}$, then , $\mathcal{OPT}_{assign}(\mathcal{T}_i) = d(h_i,c_1) + d(h_i',c_1) + d(g_i,c_2) + d(g_i',c_2)$. As a consequence of the allocation rule used by FRAC_{OE}, we have

$$d(g_i,c_1) \leq d(h_i,c_1),\ d(g_i',c_1) \leq d(h_i',c_1), d(h_i,c_2) \leq d(g_i',c_2) \text{ and } d(h_i,c_2) \leq d(h_i',c_2). \tag{3.21}$$

Furthermore,

$$\begin{aligned}
\text{FRAC}_{OE}(\mathcal{T}_i) &= d(g_i,c_1) + d(g_i',c_1) + d(h_i,c_2) + d(h_i',c_2) && (3.22)\\
&\leq d(h_i,c_1) + d(h_i',c_1) + d(g_i',c_2) + d(h_i',c_2) && (\because \text{using Equation 3.21})
\end{aligned}$$

$$\leq d(h_i, c_1) + d(h'_i, c_1) + d(g'_i, c_2) + d(h'_i, c_1) + d(c_1, c_2)$$
$$(\because \text{triangle inequality})$$
$$\leq d(h_i, c_1) + d(h'_i, c_1) + d(g'_i, c_2) + d(h'_i, c_1) + d(g_i, c_1) \tag{3.23}$$
$$+ d(g_i, c_2) \qquad (\because \text{triangle inequality})$$
$$\leq d(h_i, c_1) + d(h'_i, c_1) + d(g'_i, c_2) + d(h'_i, c_1) + d(h_i, c_1) \tag{3.24}$$
$$+ d(g_i, c_2) \qquad (\because \text{using Equation 3.21})$$
$$\leq 2d(h_i, c_1) + 2d(h'_i, c_1) + d(g_i, c_2) + d(g'_i, c_2) \tag{3.25}$$
$$\leq 2\mathcal{OPT}_{assign}(\mathcal{T}_i) \tag{3.26}$$

If there are odd number of 2-bad rounds then, let $G = (g_i, h_i)$ be the last 2-bad round. It is easy to see that $\text{FRAC}_{OE}(\mathcal{T}_i) - \mathcal{OPT}_{assign}(\mathcal{T}_i) = d(g_i, c_1) + d(h_i, c_2) - d(g_i, c_2) - d(h_i, c_1) \leq d(g_i, c_1) + d(h_i, c_2) \leq \vartheta$. Thus,

$$\text{FRAC}_{OE}(\mathcal{T}) = \begin{cases} \sum_{i=1}^{r/2} \text{FRAC}_{OE}(\mathcal{T}_i) & \text{if even no. of 2-bad rounds} \\ \sum_{i=1}^{\lfloor r/2 \rfloor} \text{FRAC}_{OE}(\mathcal{T}_i) + \vartheta & \text{Otherwise} \end{cases} \tag{3.27}$$
$$\leq 2 \sum_{i=1}^{\lfloor r/2 \rfloor} \mathcal{OPT}_{assign}(\mathcal{T}_i) + \vartheta = 2\mathcal{OPT}_{assign}(\mathcal{T}) + \vartheta \tag{3.28}$$

Here, r is the number of 2-bad rounds. and $\vartheta = 2 \sup_{x,y \in \mathcal{X}} d(x, y)$ is the diameter of the feature space. □

Corollary 3.13. *For $k = 2$ and $\tau_\ell = \frac{1}{k}$ for all $\ell \in [m]$, we have $FRAC_{OE}(\mathcal{I}) \leq \Big(2(\beta + 2)\mathcal{OPT}_{clust}(\mathcal{I}) + \vartheta\Big)$-approximate where β is approximation factor for vanilla clustering problem for any given instance $\mathcal{I}$.*

The above corollary is a direct consequence of Lemma 3.8 and the fact that $\text{FRAC}_{OE}(\hat{C}, X) \leq \text{FRAC}_{OE}(C, X)$. Here, $C, \hat{C}$ are centers of vanilla clustering and fair clustering obtained by FRAC_{OE}, respectively. The result can easily be extended for k clusters to directly obtain 2^{k-1}-approximate solution with respect to $\boldsymbol{\tau}$-ratio fair assignment problem.

Theorem 3.14. *When $\tau_\ell = \frac{1}{k}$ for all $\ell \in [m]$, an allocation returned by $FRAC_{OE}$ for given centers and data points is $\boldsymbol{\tau}$-ratio fair and satisfies 2^{k-1}-approximation guarantee with respect to an optimal $\boldsymbol{\tau}$-ratio fair assignment up to an instance-dependent additive constant.*

Proof. In the previous proof, we basically considered two length cycles. Two 1-bad allocations resulted in one type of cycles, and one 2-bad allocations resulted in another type of cycle. When the number of clusters are greater than two, then any $2 \leq q \leq k$ length cycles can be formed. Without loss of generality, let us denote $\{c_1, c_2, \ldots, c_q\}$ as the centers that are involved in forming such cycles. Further denote by set X_i^j to be the

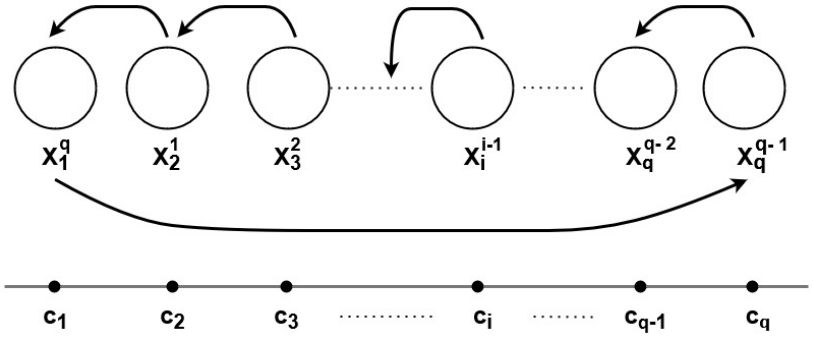


Figure 3.3: Visual representation of set X_i^j and cycle of length q for Theorem 3.14. The arrow represents the direction from the assigned center to the center in optimal clustering. Thus, for each set X_i^j we have c_i as the currently assigned center and c_j as the center in the optimal assignment.

set of data points that are allotted to cluster i by FRAC$_{OE}$ but should have been allotted to cluster j in an optimal fair clustering. The q length cycle can then be visualized in Figure 3.3 with an arrow pointing towards the optimal cluster. As the cycle is formed with respect to these data points, we have $|X_1^q| = |X_2^1| = \ldots = |X_q^{q-1}|$ The cost by FRAC$_{OE}$ algorithm is then given as:

$$\sum_{i=2}^{q} \sum_{x \in X_i^{i-1}} d(x, c_i) + \sum_{x \in X_1^q} d(x, c_1) \tag{3.29}$$

$$\leq 2 \left(\sum_{x \in X_2^1} d(x, c_1) + \sum_{x \in X_1^q} d(x, c_2) + \vartheta \right) + \sum_{i=3}^{q} \sum_{x \in X_i^{i-1}} d(x, c_i) \tag{3.30}$$

$$\leq 2(\sum_{x \in X_2^1} d(x, c_1) + \vartheta) + 2^2 \left(\sum_{x \in X_3^2} d(x, c_2) + \sum_{x \in X_1^q} d(x, c_3) + \vartheta \right) + \sum_{i=4}^{q} \sum_{x \in X_i^{i-1}} d(x, c_i) \tag{3.31}$$

$$\leq 2^{q-1} \left(\sum_{i=2}^{q} \sum_{x \in X_i^{i-1}} d(x, c_{i-1}) + \sum_{x \in X_1^q} d(x, c_q) \right) + 2^q \vartheta \tag{3.32}$$

Here, the first inequality follows by exchanging the data points in X_2^1 and X_1^q using Theorem 3.11. As the maximum length cycle possible is k, we straight away get the proof of 2^{k-1}- approximation. □

Next, in contrast with Theorem 3.14 which guarantees a 4-approximation for $k = 3$, we show that one can achieve a 2-approximation guarantee. The proof of this result relies on explicit case analysis. As the number of cases solved increases exponentially with k, one needs a better proof technique for larger values of k. We leave this analysis as an interesting future work.

Theorem 3.15. *For k=3 and $\tau_\ell = \frac{1}{k}$ allocation returned by FRAC$_{OE}$ with arbitrary centers and data points is 2-approximate with respect to optimal $\boldsymbol{\tau}$-ratio fair assignment.*

Proof. We will here find the approximation for $k = 3$ using a number of possible cases where one can have a cycle of length three. Let the centers involved in this 3-length cycle be denoted by c_i, c_j, and c_k. Note that if there is only one cycle involving these three centers, then it will lead to only constant factor approximation. The challenge is when multiple such cycles are involved. Unlike $k = 2$ proof, here we bound the cost corresponding to each cycle with respect to the cost of another cycle. The three cases shown in Figure 3.4 depicts multiple rounds when the two 3-length cycles can be formed. In the figure, if c_i is taking a data point from c_j it is denoted using an arrow from c_i to c_j. It can further be shown that it is enough to consider these three cases. Further, let $\mathcal{T}_i = \{C, \{x_i, x_j, x_k, g_i, g_j, g_k\}\}$ and $\mathcal{T}'_i = \{C, \{y_i, y_j, y_k, g'_i, g'_j, g'_k\}\}$ denote the two cycles.

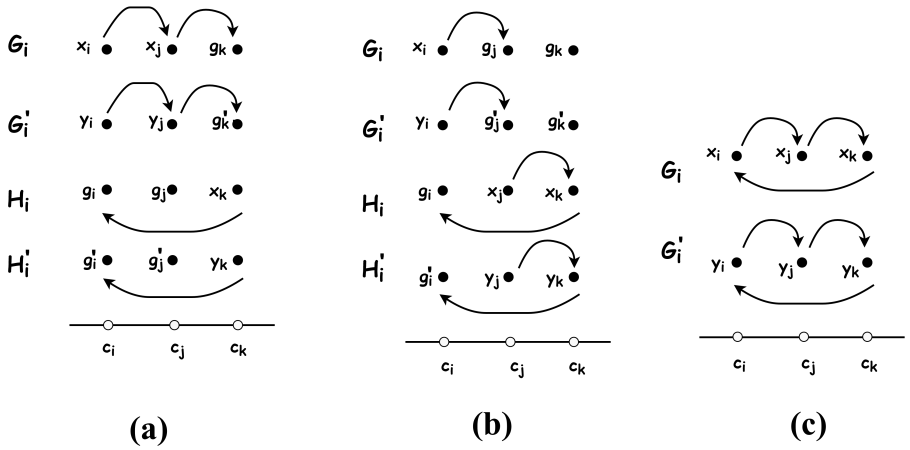


Figure 3.4: Different use cases for 3-length cycle involving k=3 clusters (a) **Case 1**: Two-three length cycle pair (G_i, H_i) and (G'_i, H'_i) (b) **Case 2**: Second possibility of two-three length cycle pair (G_i, H_i) and (G'_i, H'_i) (c) **Case 3**: Three length cycle pair (G_i, G'_i).

Case 1: In this case, we bound the rounds shown in Figure 3.4(a). Let, one cycle completes in rounds G_i, H_i (i.e. using points from $\mathcal{T}_i$) and another cycle completes in rounds G'_i, H'_i (using points from $\mathcal{T}'_i$). Then,

$$\mathcal{OPT}_{assign}(\mathcal{T}_i) = d(x_i, c_j) + d(x_j, c_k) + d(g_k, c_k) + d(g_i, c_i) + d(g_j, c_j) + d(x_k, c_i) \quad (3.33)$$

$$\mathcal{OPT}_{assign}(\mathcal{T}'_i) = d(y_i, c_j) + d(y_j, c_k) + d(g'_k, c_k) + d(g'_i, c_i) + d(g'_j, c_j) + d(y_k, c_i) \quad (3.34)$$

Further,

$$\text{FRAC}_{OE}(\mathcal{T}_i) = d(x_i, c_i) + d(x_j, c_j) + d(g_k, c_k) + d(g_i, c_i) + d(g_j, c_j) + d(x_k, c_k) \quad (3.35)$$

$$\leq d(g'_i, c_i) + d(g'_j, c_j) + d(g_k, c_k) + d(g_i, c_i) + d(g_j, c_j) + d(x_k, c_k) \quad (3.36)$$

Now,

$$d(x_k, c_k) \leq d(x_k, c_i) + d(c_i, c_k) \leq d(x_k, c_i) + d(c_i, c_j) + d(c_j, c_k) \quad (3.37)$$

$$\leq d(x_k, c_i) + d(x_i, c_i) + d(x_i, c_j) + d(x_j, c_j) + d(x_j, c_k) \quad (3.38)$$

$$\leq d(x_k, c_i) + d(y_k, c_i) + d(x_i, c_j) + d(y_i, c_j) + d(x_j, c_k) \quad (3.39)$$

Combining the above two, we get:

$$\text{FRAC}_{OE}(\mathcal{T}_i) \leq \mathcal{OPT}_{assign}(\mathcal{T}_i) + \mathcal{OPT}_{assign}(\mathcal{T}_i') \tag{3.40}$$

Thus, the cost of each cycle can be bounded by the sum of the optimal cost of its own and the optimal cost of the next cycle. If we take sum over all such cycles, we will get 2-approximation result plus a constant due to the last remaining cycle.

Case 2: In this case, we bound the rounds shown in Figure 3.4(b). The optimal assignments will be

$$\mathcal{OPT}_{assign}(\mathcal{T}_i) = d(x_i, c_j) + d(g_j, c_j) + d(g_k, c_k) + d(g_i, c_i) + d(x_j, c_k) + d(x_k, c_i) \tag{3.41}$$
$$\mathcal{OPT}_{assign}(\mathcal{T}_i^{'}) = d(y_i, c_j) + d(g_j', c_j) + d(g_k', c_k) + d(g_i', c_i) + d(y_j, c_k) + d(y_k, c_i) \tag{3.42}$$

Also, we know that

$$\begin{aligned}
\text{FRAC}_{OE}(\mathcal{T}_i) &= d(x_i, c_i) + d(g_j, c_j) + d(g_k, c_k) + d(g_i, c_i) + d(x_j, c_j) + d(x_k, c_k) && (3.43)\\
&\leq d(g_i', c_i) + d(g_j, c_j) + d(g_k, c_k) + d(g_i, c_i) + d(y_k, c_j) + d(x_k, c_k) && (3.44)\\
&\leq d(g_i', c_i) + d(g_j, c_j) + d(g_k, c_k) + d(g_i, c_i) + d(y_k, c_j) + d(y_j, c_k) && (3.45)\\
&\leq d(g_i', c_i) + d(g_j, c_j) + d(g_k, c_k) + d(g_i, c_i) + d(y_k, c_i) + d(c_i, c_j) + && (3.46)\\
& d(y_j, c_k) && (3.47)\\
&\leq d(g_i', c_i) + d(g_j, c_j) + d(g_k, c_k) + d(g_i, c_i) + d(y_k, c_i) + d(x_i, c_i) + && (3.48)\\
& d(x_i, c_j) + d(y_j, c_k) && (3.49)\\
&\leq d(g_i', c_i) + d(g_j, c_j) + d(g_k, c_k) + d(g_i, c_i) + d(y_k, c_i) + d(x_k, c_i) + && (3.50)\\
& d(x_i, c_j) + d(y_j, c_k) && (3.51)\\
&\leq d(g_i', c_i) + d(g_j, c_j) + d(g_k, c_k) + d(g_i, c_i) + d(y_k, c_i) + d(x_k, c_i) + && (3.52)\\
& d(x_i, c_j) + d(y_j, c_k) + d(x_j, c_k) + d(y_i, c_j) + d(g_j', c_j) + d(g_k', c_k) && (3.53)
\end{aligned}$$

Combining the above two, we get:

$$\text{FRAC}_{OE}(\mathcal{T}_i) \leq \mathcal{OPT}_{assign}(\mathcal{T}_i) + \mathcal{OPT}_{assign}(\mathcal{T}_i') \tag{3.54}$$

Case 3: Here again, we will have two allocation rounds, namely $G_i, G_i^{'}$ as shown in Figure 3.4 (c). It is easy to see that for this case,

$$\text{FRAC}_{OE}(G_i) \leq \mathcal{OPT}_{assign}(\mathcal{T}_i') \tag{3.55}$$

This completes the proof for $k = 3$. □

The following proposition proves that 2-approximation guarantee is tight with respect to the FRAC_{OE} algorithm.

Proposition 3.16. *There is an instance with arbitrary centers and data points on which*

FRAC$_{OE}$ achieves 2-approximation with respect to the optimal assignment.

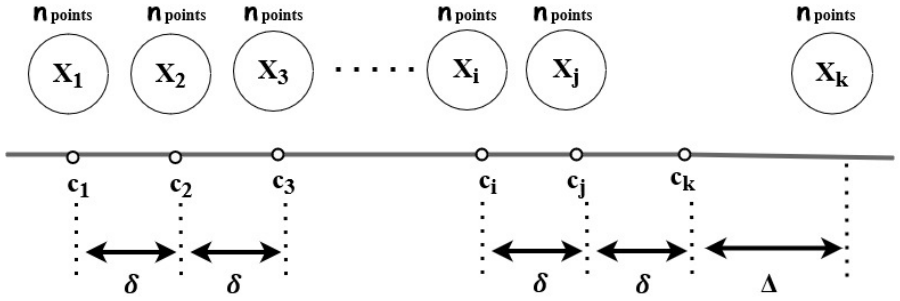


Figure 3.5: The worst case example for fair clustering instance.

Proof. The worst case for any fair clustering instance can be the situation wherein rather than choosing the data points from the center's own set of optimal data points, it prefers data points from other centers. One such example is depicted in Figure 3.5. In this example, we consider k centers. For each of these centers, we have a set of n optimal data points that are at a negligible distance (say zero), and these sets are denoted by X_i for center c_i except the last center c_k. The set of optimal data points for center c_k is located at a distance Δ such that $\Delta = (k-1)\delta$ where δ is the distance between all the centers. Now, we will approximate the tightest bound on the cost. In the optimal assignment, each cluster center will take data points from its optimal set of data points. Thus, the optimal cost can be summed up as

$$\mathcal{OPT}_{assign} = \sum_{x_i \in X_1} d(x_i, c_1) + \sum_{x_i \in X_2} d(x_i, c_2) + \ldots + \sum_{x_i \in X_k} d(x_i, c_k) \tag{3.56}$$

$$= 0 + 0 + 0 + n\Delta \tag{3.57}$$

If one uses round-robin based FRAC$_{OE}$ to solve an assignment problem, then at the start of $t = 0^{th}$ round, each of the set X_i has n data points. Now since Δ is quite large as compared to δ so c_k will prefer to choose data points from the set of previous center c_{k-1}. The remaining centers will take data points from their respective set of optimal data points as those data points will have the least cost. This type of assignment will continue until all the data points in set X_{k-1} get exhausted. Thus, the cost after $n/2$ rounds will be

$$Cost_1 = \sum_{x_i \in X_1} d(x_i, c_1) + \ldots + \sum_{x_i \in X_{k-1}} d(x_i, c_{k-1}) + \sum_{x_i \in X_{k-1}} d(x_i, c_k) \tag{3.58}$$

$$= 0 + 0 + 0 + \frac{n\delta}{2} \tag{3.59}$$

Now, as all the data points in set X_{k-1} are exhausted, both c_{k-1} and c_k will prefer to choose the data points from set X_{k-2}. The other centers will still continue to choose the data points from their respective optimal sets. It should be noted that now $\frac{n}{2}$ data points are left with the center X_{k-2} that are being distributed amongst 3 clusters. Such

assignments will take place for the next $\frac{n}{6}$ rounds, and after that, the set X_{k-2} will get exhausted. The cost incurred to different centers in such an assignment will be

$$Cost_2 = \sum_{x_i \in X_1} d(x_i, c_1) + \ldots + \sum_{x_i \in X_{k-2}} d(x_i, c_{k-2}) + \sum_{x_i \in X_{k-2}} d(x_i, c_{k-1}) \tag{3.60}$$

$$+ \sum_{x_i \in X_{k-2}} d(x_i, c_k) \tag{3.61}$$

$$= \frac{n\delta}{6} + \frac{2n\delta}{6} \tag{3.62}$$

$$= \frac{3n\delta}{6} = \frac{n\delta}{2} \tag{3.63}$$

It is easy to see that the additional cost that is incurred at each phase will be $\frac{n\delta}{2}$ until the only left-out data points are from X_k. The total number of such phases will be $k-1$. Thus, exhibiting a cost of $\frac{n(k-1)\delta}{2}$. Further, at the last round all the data points from X_k need to be equally distributed amongst $X_1, X_2, \ldots, X_k$, incurring the total cost of $((k-1)\delta + \Delta + (k-2)\delta + \Delta + \ldots + \delta + \Delta + \Delta)\frac{n}{k}$. Thus, the total cost by FRAC_{OE} is given as:

$$Cost_{\text{FRAC}_{OE}} = \frac{n(k-1)\delta}{2} + ((k-1)\delta + \Delta + (k-2)\delta + \Delta + \ldots + \delta + \Delta + \Delta)\frac{n}{k} \tag{3.64}$$

$$= \frac{n(k-1)\delta}{2} + \frac{nk(k-1)\delta}{2k} + \frac{nk\Delta}{k} \tag{3.65}$$

$$= n(k-1)\delta + n\Delta \tag{3.66}$$

$$= 2n\Delta \tag{3.67}$$

□

Research gap: Theorem 3.14 suggests that the approximation ratio with respect to the number of clusters k can be exponentially bad. However, our experiments show—agreeing with our finding on small values of $k(\leq 3)$—that the performance of FRAC_{OE} does not degrade with k. To assert a 2-approximation bound for general k, a novel proof technique is needed, and we leave this analysis as an interesting future work. We conclude with the following conjecture.

Conjecture 3.17. *$FRAC_{OE}$ is 2-approximate with respect to optimal $\boldsymbol{\tau}$-ratio fair assignment problem for any value of k.*

We note that FRAC_{OE} uses vanilla k-means/k-median algorithm followed by one round of fair assignment procedure. It is left to show that the output of the returned by the FRAC_{OE} algorithm indeed converges to approximately optimal $\boldsymbol{\tau}$-ratio allocation in finite time. Convergence guarantees of vanilla clustering algorithms are well known in the literature ([182, 183, 184]). As a fair assignment procedure, it performs corrections for all available data points only once. Thus, FRAC_{OE} is bound to converge. This gives us the following lemma.

Lemma 3.18. *$FRAC_{OE}$ algorithm converges.*

3.5.2 Guarantees for FRAC$_{OE}$ for general τ

Given an instance $\mathcal{T}$, centers C, and set of data points X, we start with a simple observation that problem of solving $\boldsymbol{\tau}$-ratio fair assignment can be divided into two subproblems:

1. Solving optimal $1/k$-ratio fair assignment problem on subset of data points $X_1 \in X$ such that $|X_1| = \sum_{\ell \in [m]} k\tau_\ell n_\ell$.
2. Solving optimal fair assignment problem on $X_2 \in X \setminus X_1$ without any fairness constraint.

Let us denote the first instance by $\mathcal{T}^{1/k}$ and second instance with $\mathcal{T}^0$, i.e. $\mathcal{T}^{1/k} = \{X_1, C\}$ and $\mathcal{T}^0 = \{X_2, C\}$.

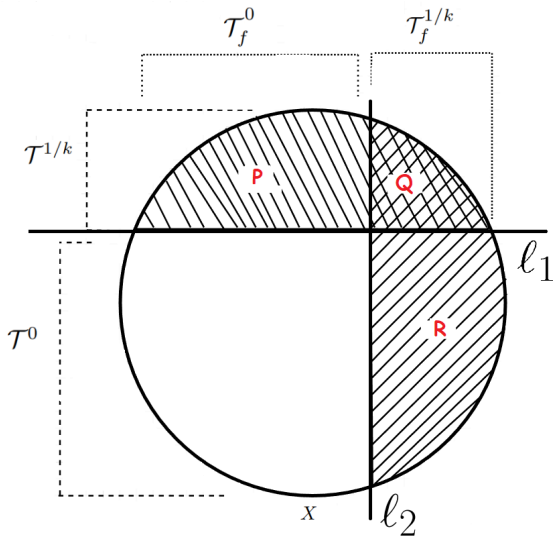


Figure 3.6: Set of data points X divided into instance $\mathcal{T}^{1/k}$ and $\mathcal{T}^0$. Further the instances $\mathcal{T}_f^{1/k}$ and $\mathcal{T}_f^0$ are depicted in the same set of data points X leading to formation of regions P, Q, R.

Lemma 3.19. *There exist two separate instances $\mathcal{T}^{1/k}$ with τ=$\{1/k\}_{\ell=1}^m$ and $\mathcal{T}^0$ with τ=$\{0\}_{\ell=1}^m$ such that solving the fair assignment problem on instance $\mathcal{T}$ can be divided into solving fair assignment on these two instances, i.e., $\mathcal{OPT}_{assign}(\mathcal{T}) = \mathcal{OPT}_{assign}(\mathcal{T}^{1/k}) + \mathcal{OPT}_{assign}(\mathcal{T}^0)$.*

Proof. The $\mathcal{T}$ instance requires that each cluster should have at least $\tau_\ell n_\ell$ number of data points for each protected group value. The remaining data points can be allocated in an optimal manner without any fairness constraint. Therefore in an optimal assignment, there exists a set X_1^{OPT} such that $|X_1^{OPT}| = \sum_{\ell=1}^m \tau_\ell n_\ell k$ that satisfies the $\tau-$ratio fairness with $\tau_\ell = 1/k \ \forall \ell \in [m]$. □

Let X_1^f be the set of data points that are allocated in line number 4 by Algorithm 2. Further, let $\mathcal{T}_f^{1/k}$ be an instance to τ-ratio fair assignment problem with $\tau = \{1/k\}_{\ell=1}^m$ and consisting of data points X_1^f and $\mathcal{T}_f^0$ be instance when τ=$\{0\}_{\ell=1}^m$ by FRAC$_{OE}$ (depicted

in Figure 3.6). Then, our next lemma shows that the partition returned by FRAC_{OE} is the optimal one.

Lemma 3.20. *$\mathcal{OPT}_{assign}(\mathcal{T}_f^{1/k}) + \mathcal{OPT}_{assign}(\mathcal{T}_f^0) \leq \mathcal{OPT}_{assign}(\mathcal{T}^{1/k}) + \mathcal{OPT}_{assign}(\mathcal{T}^0)$ for any partition $\mathcal{T}^{1/k}$ and $\mathcal{T}^0$. Thus, $\mathcal{OPT}_{assign}(\mathcal{T}) = \mathcal{OPT}_{assign}(\mathcal{T}_f^{1/k}) + \mathcal{OPT}_{assign}(\mathcal{T}_f^0)$.*

Proof. Let optimal fair assignment on the set of data points X create a partition along the axis given by line ℓ_1 in Figure 3.6. This partition gives us two set of instances $\mathcal{T}^{1/k}$, $\mathcal{T}^0$ (as described earlier). Further, FRAC_{OE} achieves a partition along axis given by line ℓ_2 denoted by $\mathcal{T}_f^{1/k}$, $\mathcal{T}_f^0$. Now region Q contains the data points in the overlap of $\mathcal{T}^{1/k}$ and $\mathcal{T}_f^{1/k}$. As we are talking about the optimal assignment problem, these data points will be assigned to the same centers and hence we can ignore these data points for further analysis. Let the data points allocated to any center c_j in $\mathcal{T}_f^{1/k}$ by FRAC_{OE} in set R be $R_j = \{x_1, x_2, x_3, \ldots, x_{m_j}\}$ and data points allocated to c_j in partition P be $P_j = \{y_1, y_2, y_3, \ldots, y_{m_j}\}$. Also, let $g : R_j \to P_j$ be a mapping function that maps any data point x_i assigned to center j with $\mathcal{T}_f^{1/k}$ to some data point y_i assigned to same center when partition under consideration is $\mathcal{T}^{1/k}$. Then, we have $\mathcal{OPT}_{assign}(\mathcal{T}_f^{1/k}) \leq \text{FRAC}_{OE}(\mathcal{T}_f^{1/k}) = \sum_{j=1}^{k}\sum_{i=1}^{m_j} d(x_i, c_j) \leq \sum_{j=1}^{k}\sum_{i=1}^{m_j} d(y_i, c_j) = \mathcal{OPT}_{assign}(\mathcal{T}^{1/k})$. This is because for each $x_i \in R_j, \exists y_j \in P_j$ such that despite point y_i being available to center c_j, it chose the point x_i. As other data points have no such constraint, we have, $\mathcal{OPT}_{assign}(\mathcal{T}_f^0) \leq \mathcal{OPT}_{assign}(\mathcal{T}^0)$.

□

Theorem 3.21. *For k=2, 3 and any general τ vector, an allocation returned by $FRAC_{OE}$ guarantees $\boldsymbol{\tau}$-ratio fairness and satisfies $(2(\beta+2)\mathcal{OPT}_{clust})$-approximate guarantee with respect to a fair clustering problem where β is approximation factor for the vanilla clustering problem.*

Proof. With the help of Lemma 3.19 the cost of FRAC_{OE} on instance $\mathcal{T}_f$ can be computed as,

$$\text{FRAC}_{OE}(\mathcal{T}) = \text{FRAC}_{OE}(\mathcal{T}_f^{1/k}) + \text{FRAC}_{OE}(\mathcal{T}_f^0) \tag{3.68}$$

Now, from previous Section 3.5.1, $\text{FRAC}_{OE}(\mathcal{T}_f^{1/k}) \leq 2\mathcal{OPT}_{assign}(\mathcal{T}_f^{1/k})$.
Also, as $\mathcal{T}_f^0$ is solved for τ=$\{0\}_{\ell=1}^m$ i.e. assignment is carried solely on the basis of vanilla clustering (k-means/k-median), we have $\text{FRAC}_{OE}(\mathcal{T}_f^0) = \mathcal{OPT}_{assign}(\mathcal{T}_f^0) \leq 2\mathcal{OPT}_{assign}(\mathcal{T}_f^0)$.
Equation 3.68 becomes,

$$\begin{aligned}
\text{FRAC}_{OE}(\mathcal{T}) &\leq 2\mathcal{OPT}_{assign}(\mathcal{T}_f^{1/k}) + 2\mathcal{OPT}_{assign}(\mathcal{T}_f^0) && (3.69)\\
&\leq 2\mathcal{OPT}_{assign}(\mathcal{T}) && \text{(using Lemma 3.19)}\\
&\leq 2(\beta+2)\mathcal{OPT}_{clust}(\mathcal{I}) && \text{(using Lemma 3.8)}
\end{aligned}$$

□

3.6 Fair Round Robin Algorithm for Clustering (FRAC) –A Heuristic Approach

We now propose another algorithm, a general version of FRAC_{OE} where the fairness constraints are satisfied at each allocation round: Fair Round-Robin Algorithm for Clustering FRAC (described in Algorithm 3). FRAC runs a fair assignment problem at each iteration of a vanilla clustering algorithm. This may lead to the shuffling of data points, affecting the position of next-step cluster centers. Also, modifying allocation does not preserve the convergence guarantee of the vanilla clustering algorithm.

Algorithm 3: τ-FRAC

Input: set of data points X, number of clusters k, fairness requirement vector $\boldsymbol{\tau}$, range of protected group m, clustering objective norm p

Output: cluster centers C and assignment function ϕ

Choose the random centers as C
while *UntilConvergence* **do**
  for *each* $x_i \in X$ **do**
    $\phi(x_i) = \text{argmin}_m \, d(x_i, c_m)$
  end
  $(C, \phi) = \textsc{FairAssignment}(C, X, k, \tau, m, p, \phi)$
end

Therefore, it is theoretically hard to analyze FRAC as it is an in-processing algorithm, and each round's allocation depends upon previous rounds, i.e., the rounds are not independent. However, in experiments, we see that FRAC performs better than FRAC_{OE}, and baseline methods on a wide range of real-world datasets. We experimentally show the convergence of both FRAC and FRAC_{OE} on real-world datasets. These empirical results suggest that either the worst-case instances for FRAC are unrealistic or a significantly different proof technique is needed to show the convergence guarantee. We leave this as an interesting future direction. As both FRAC_{OE} and FRAC solve the fair assignment problem on top of the vanilla clustering problem, one can use them to find fair clustering for center-based approaches, i.e., k-means and k-median.

3.7 Experimental Result and Discussion

We validate the performance of the proposed algorithms against state-of-the-art (`SOTA`) approaches across many benchmark datasets listed below:

- **Adult**[4] **(Census)**- The data set contains information of 32562 individuals from the 1994 census, of which 21790 are males and 10771 are females. We choose five groups as feature set: age, fnlwgt, education_num, capital_gain, hours_per_week. The binary-valued protected group is sex, which is consistent with prior literature [42, 12, 11, 10]. The *dataset ratio* is 0.49.

[4]https://archive.ics.uci.edu/ml/datasets/Adult

- **Bank**[5]- The dataset consists of marketing campaign data of a Portuguese bank. It has data of 41108 individuals, of which 24928 are married, 11568 are single, and 4612 are divorced. We choose six groups as the feature set: age, duration, campaign, cons.price.idx, euribor3m, nr.employed. The ternary-valued feature 'martial status' is chosen as the protected group to be consistent with prior literature, resulting in a Balance of 0.18 [42, 12, 11, 10].

- **Diabetes**[6]- The dataset contains clinical records of 130 US hospitals over ten years. There are 54708 and 47055 hospital records of males and females, respectively. Consistent with the prior literature, only two features: age, time_in_hospital are used for the study [42]. Gender is treated as the binary-valued protected group yielding a Balance of 0.86.

- **Census II**[7]- It is the largest dataset used in this study containing 2458285 records from of US 1990 census, out of which 1191601 are males, and 1266684 are females. We chose 24 groups commonly used in prior literature for this study [12, 10]. Sex is the binary-valued protected group. The *dataset ratio* is 0.94.

Dataset Name	#Cardinality	#Feature Attribute	Protected Group	Protected Group Cardinality	Protected Group Composition			Dataset Ratio
Adult (Census)	32562	5	gender	binary	21790 males	10771 females	–	0.49
Bank	41108	6	marital status	ternary	24928 married	11568 unmarried	4612 divorced	0.18
Diabetes	101763	2	gender	binary	54708 males	47055 females	–	0.86
Census II	2458285	24	gender	binary	1191601 males	1266684 females	–	0.94

Table 3.1: Characteristics for real-world datasets commonly used in the evaluation of fair clustering algorithms. Number of feature groups excludes protected group and for complete list of feature groups see Section 3.7.

The dataset characteristics are summarized in Table 3.1. We compare the application of FRAC to k-means and k-median against the following baseline and SOTA approaches

- **Vanilla k-means**: An Euclidean distance-based k-means algorithm that does not incorporate fairness constraints

- **Vanilla k-median**: An Euclidean distance-based k-median algorithm that does not incorporate fairness constraints.

- **Bera et al. [12]**: The approach solves the fair clustering problem through an LP formulation. Fairness is added as an additional constraint in the LP by bounding

[5]https://archive.ics.uci.edu/ml/datasets/Bank+Marketing
[6]https://archive.ics.uci.edu/ml/datasets/Diabetes+130-US+hospitals+for+years+1999-2008
[7]https://archive.ics.uci.edu/ml/datasets/US+Census+Data+%281990%29

the minimum (minority protection see Definition 3.3) and maximum (restricted dominance see Definition 3.4) fraction of data points belonging to the particular protected group in each cluster. Due to the high computational complexity of the k-median version of the approach, we restrict the comparison to the k-means version. Furthermore, the algorithm fails to converge within a reasonable amount of time when the number of clusters is greater than 10 for larger datasets.

- **Ziko et al. [10]**: This approach formulates a regularized optimization function incorporating clustering objective and fairness error. It does not allow the user to give an arbitrary fairness guarantee but computes the optimal trade-off by tuning a hyper-parameter λ. We compare against both the k-means and k-median versions of the algorithm. We observed that the hyper-parameter λ is extremely sensitive to the datasets and the number of clusters. Further, tuning this hyper-parameter is computationally expensive. We were able to tune the value of λ in a reasonable amount of time only for adult and bank datasets for k-means clustering and a varying number of clusters. Due to the added complexity of k-medians, we were able to fine-tune λ only for the adult dataset. For the other cases, we have used the hyper-parameter value reported by Ziko et al. (we refer to this as Ziko et al. (untuned) version). We have used the same value across varying numbers of cluster centers. The paper does not report any results for the diabetes dataset; we have chosen the best λ value over a single run of fine-tuning. This value is used across all experiments related to the diabetes dataset.

- **Backurs et al. [11]**: This approach computes the fair clusters using fairlets in an efficient manner and is the extension of Chierichetti et al. [42]. This approach can only be integrated with k-median clustering. Further, we could not compare against k-median version of Chierichetti et al. [42] due to high computational ($O(n^2)$) and space complexities. We offset this comparison using Backurs et al. [11] that gives us better performance than Chierichetti et al. [42].

We use the following popular metrics in the literature for measuring the performance of the different approaches.

- **Objective Cost**: We use the squared Euclidean distance ($p = 2$) as the objective cost to estimate the cluster's compactness (see Definition 3.1).

- **Balance**: The Balance is calculated using Definition 3.2.

- **Fairness Error** [10]: It is the Kullback-Leibler (KL) divergence between the required protected group proportion $\boldsymbol{\tau}$ and achieved proportion within the clusters:

$$FE(\mathcal{C}) = \sum_{C_j \in \mathcal{C}} \sum_{\ell \in [m]} \left(-\tau_\ell \log \left(\frac{q_\ell}{\tau_\ell} \right) \right) \textit{where } q_\ell = \left(\frac{\sum_{x_i \in C_j} \mathbb{I}(\rho(x_i) = \ell)}{\sum_{x_i \in X} \mathbb{I}(\rho(x_i) = \ell)} \right) \tag{3.70}$$

The $\boldsymbol{\tau}$ vector in fairness error captures the target proportion in each cluster for different protected groups $\ell \in [m]$. It is similar to the input vector $\boldsymbol{\tau}$ for FRAC and FRAC$_{OE}$. In Bera et al. [12], the target vector is denoted by δ (refer Section 3.7.3 for details on the parameter δ). We report the average and standard deviation of the performance measures across 10 independent trials for every approach. The code for all the experiments is publicly available[8]. We begin the empirical analysis of various approaches under both k-means and k-median settings for a fixed value of k (=10) in line with the previous literature. The top and bottom rows in Figure 3.7 summarize the results obtained for the k-means and k-median settings, respectively.

Observation for $k-$means:

- Ziko et al. [10] achieves the lowest objective cost but with poor performance on both the fairness measures.
- FRAC and FRAC$_{OE}$ achieves maximum Balance and zero fairness error with significantly lower objective cost compared to Bera et al. [12].

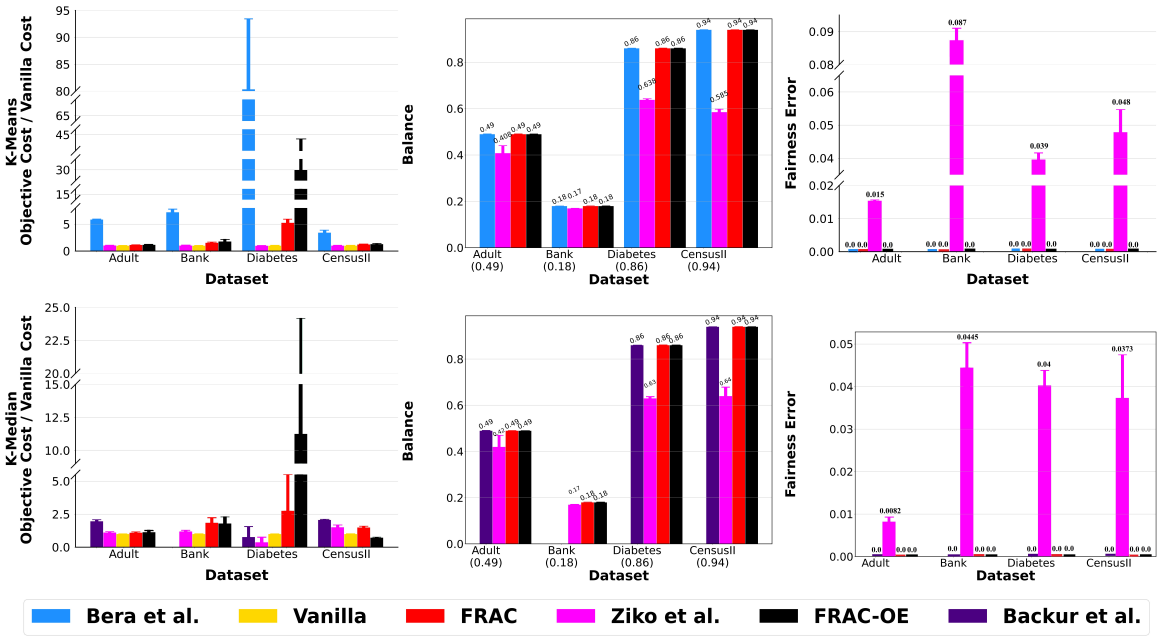


Figure 3.7: The plot in the first row shows the variation in evaluation metrics for k=10 clusters. The objective cost is scaled against vanilla objective cost. For Ziko et al., the λ values for k-means and k-median are taken to be the same as in their paper. The second row comprises plots for k-median setting on the same k value. It should be noted that Backurs et al. do not work for bank dataset which has a ternary valued protected group. The target Balance of each dataset is evident from the axes of the plot. (Best viewed in color).

Observations for $k-$median setting:

- Backurs et al. [11] results in fair clusters with high objective cost.

[8] https://github.com/shivi98g/Fair-k-means-Clustering-via-Algorithmic-Fairness

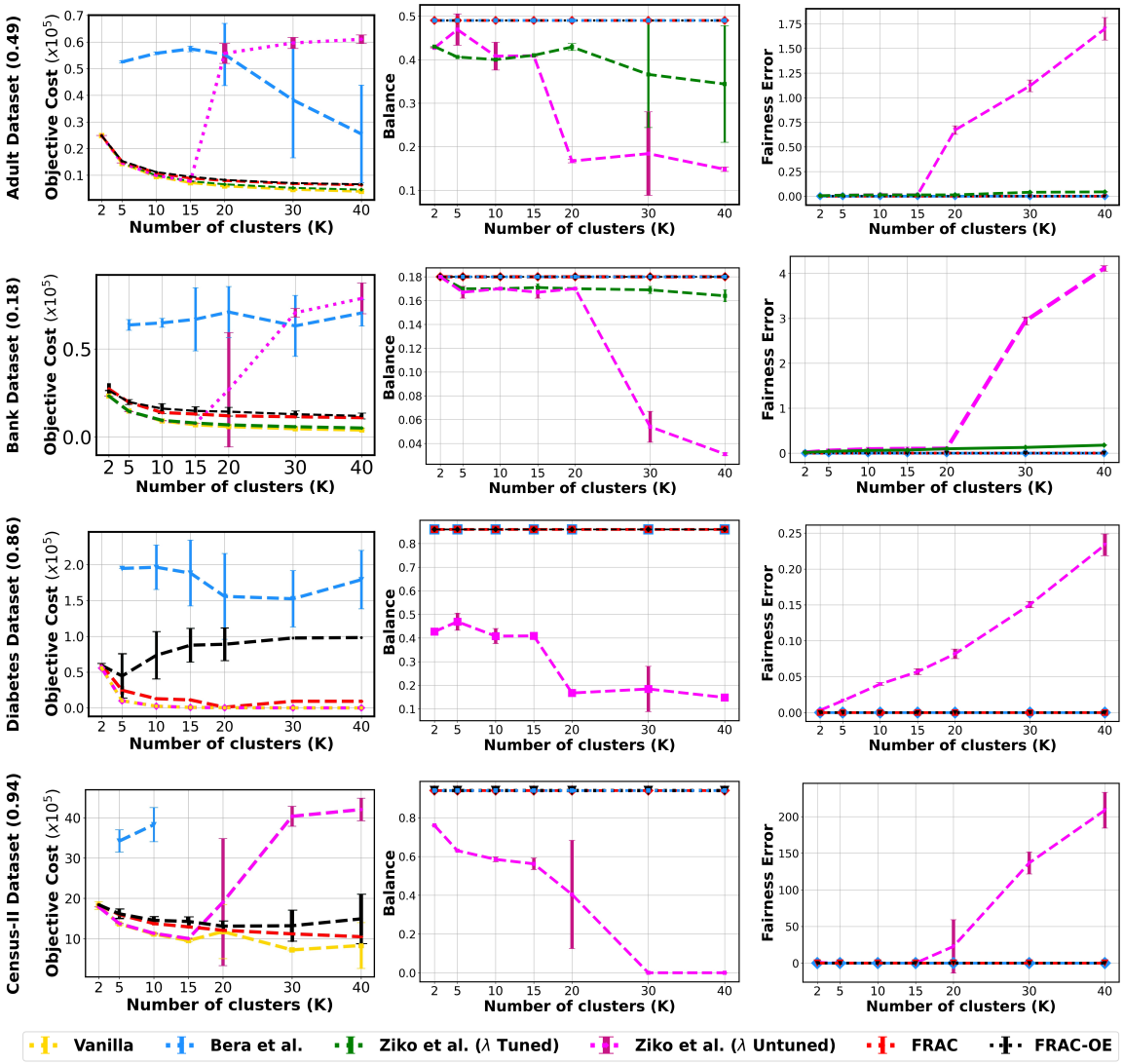


Figure 3.8: The line plot shows the variation of evaluation metrics over a varying number of cluster centers for k-means setting. The hyper-tuned variation of Ziko et al. is available only for adult and bank datasets due to expensive computational requirements. For other datasets, the hyper-parameter λ is taken the same as that is reported in Ziko et al. paper, i.e. λ=9000, 6000, 6000, 500000 for Adult, Bank, Diabetes and Census II datasets respectively. For similar reasons, Bera et al. results for Census-II are evaluated for k=5 and k=10. (Best viewed in color).

- Ziko et al. [10] achieves better objective costs trading off for fairness.
- FRAC and FRAC$_{OE}$ obtain the least fairness error and a Balance that is equal to the required dataset ratio ($\tau_\ell = \frac{1}{k}$) while having comparable objective cost.

3.7.1 Comparison across a Varying Number of Clusters (k)

In this experiment, we measure the performance of the k-means version of the different approaches across all the datasets as the number of clusters increases from 2 to 40. Figure 3.8 summarizes the results obtained for 2, 5, 10, 15, 20, 30, and 40 number of clusters on all datasets. For the largest dataset, Census-II, results are obtained for only $k = 5$ and

$k = 10$ due to the large time complexity of solving the LP problem.

Observations:

- Bera et al. [12] maintains fairness but with a much higher objective cost and fails to return any solution for $k = 2$.
- Ziko et al. (tuned) objective cost is close to vanilla k-means on the Adult and Bank datasets but at significant deterioration in fairness metrics.
- Ziko et al. (untuned) has high objective cost as well as fairness error indicating the sensitivity to the hyper-parameter λ.
- FRAC gives the best result maintaining a relatively low objective cost without compromising fairness.
- FRAC$_{OE}$ has a marginal cost difference from FRAC with the same fairness guarantees showing its efficacy.
- Theoretically FRAC$_{OE}$ show approximation factor of 2^{k-1} but the experimental performance does not degrade with increase in k. This validates our conjecture.

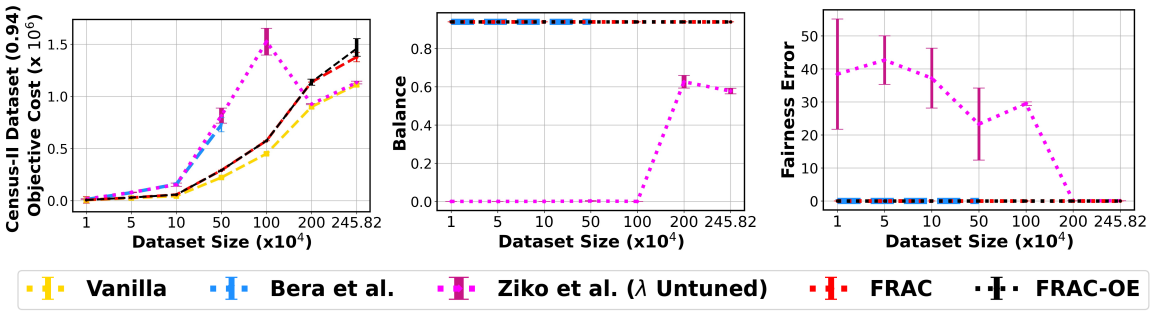


Figure 3.9: The line plot shows the variation of evaluation metrics over varying data set size for k(=10)-means setting. The hyper-parameter λ=500K is taken the same as that is reported in the Ziko et al. paper for the Census-II dataset due to expensive computational requirements. For similar reasons, Bera et al.'s results for Census-II are evaluated at up to 500K. The target balance for Census-II is evident from plot axes, and the complete data set size is 245.82×10^4. (Best viewed in color).

3.7.2 Comparison across Varying Data Set Sizes

In this experiment, we measure the performance of k(=10)-means the version of different approaches as the number of data points in the dataset increases. For this experiment, we use the largest data set, Census-II. Figure 3.9 shows the plots for evaluation metrics on varying dataset sizes increasing from 10000 to a complete size of 2458285 data points. Due to the high computation requirements for Bera et al. [12] (refer Section 3.7.4), we limit the results up to $500k$ data points. For Ziko et al., owning to high tuning time (refer run time analysis section 3.7.4), we use the hyper-parameter value for Census-II reported in Ziko et al. i.e. λ=500000 for complete dataset.

Observations:

- Bera et al. maintains strict fairness at higher objective costs.
- Initially, objective cost in Ziko et al. increases with dataset size but decreases on larger sizes (sensitive to hyper-parameter), but at significant degradation in fairness metrics.
- FRAC and FRAC$_{OE}$ achieve strict fairness guarantees with a slight increase in objective cost from vanilla clustering.

3.7.3 Additional Analysis on Proposed Algorithms

In this section, we perform additional studies on FRAC and FRAC$_{OE}$ to illustrate their effectiveness.

FRAC vs FRAC$_{OE}$

FRAC uses round-robin allocation after every clustering iteration. On the contrary, FRAC$_{OE}$ applies the round-robin allocation only at the end of clustering. Both approaches will result in a fair allocation but might exhibit different objective costs. We conduct an experiment under the $k(=10)$-means setting to study the difference in the objective costs for the two approaches. Like other experiments, we conduct this experiment over ten independent runs and plot the mean objective cost (line) and standard deviation (shaded region) at each iteration over different runs.
Observations: The plots in Figure 3.10 indicates that FRAC has a lower objective cost at convergence than FRAC$_{OE}$. The plot for FRAC$_{OE}$ follows the same cost variation as that of vanilla k-means in the initial phase, but at the end there is a sudden jump that overshoots the cost of FRAC (to accommodate fairness constraints). Thus, applying fairness constraints after every iteration is better than applying it only once at the end. The plot also helps us experimentally visualize the convergence of both FRAC and FRAC$_{OE}$ algorithms. It may be observed that the change in objective cost becomes negligible after a certain number of iterations.

Impact of order in which the centers pick the data points

FRAC assumes an arbitrary order of the centers for allocating data points at every iteration. We verify if the order in which the centers pick the data points impacts the objective cost. We vary the order of the centers picking the data points for the $k(=10)$-means clustering version. We report the objective cost variance computed across 100 permutations of the ten centers. Applying the permutations at every iteration in FRAC is an expensive proposition. Hence, we restrict the experiment to the FRAC$_{OE}$ version. The variance of the 100 final converged objective costs (averaged over ten trials) is presented in Figure 3.11 (a).

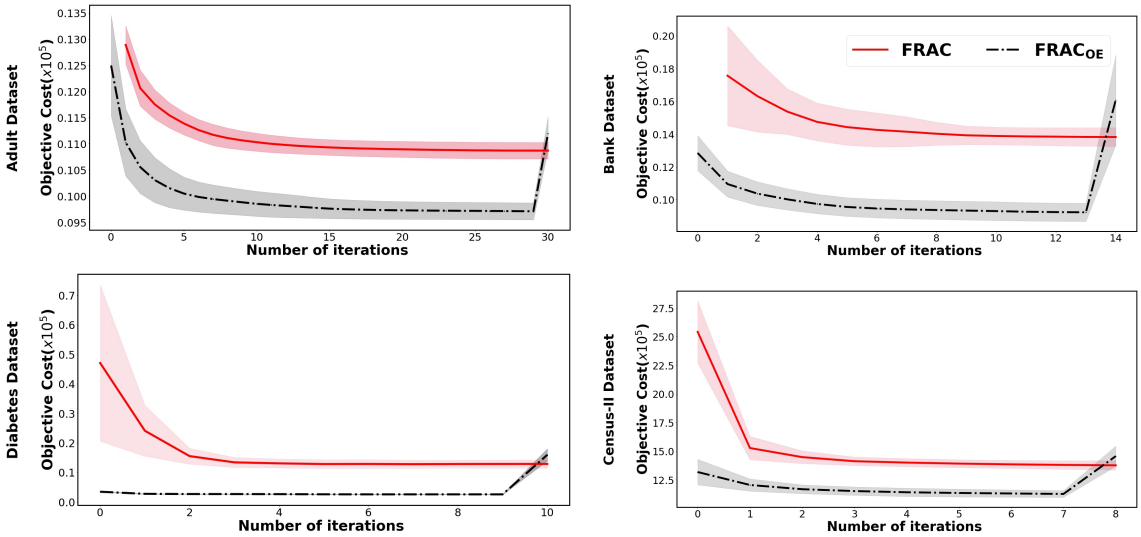
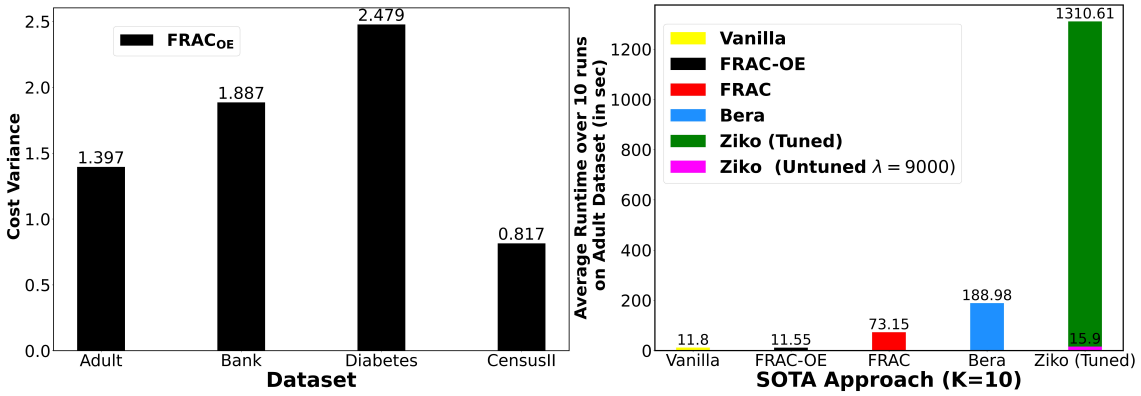


Figure 3.10: The cost variation over the iterations for different approaches in k(=10)-means.



(a) (b)

Figure 3.11: (a) Bar plot shows the variance in objective cost over different 100 random permutations of converged centers returned by vanilla k-means clustering in FRAC$_{OE}$. (b) k-means runtime analysis of different SOTA approaches on the Adult dataset for k=10.

Observations: It is evident from the plot that the variance is consistently extremely small for all datasets. Thus, we conclude that FRAC$_{OE}$ (and FRAC by extension) is invariant to the order in which the centers pick the data points.

Comparison for $\boldsymbol{\tau}$-ratio on fixed number of clusters(k)

All the experiments till now considered the Balance to be the same as the dataset ratio ($\tau_\ell = \frac{1}{k}$). But FRAC and FRAC$_{OE}$ can be used to obtain any desired $\boldsymbol{\tau}$-ratio fairness constraints other than *dataset ratio*. The results for other $\boldsymbol{\tau}$ vector values on k=10 number of clusters are reported in Table 3.2. We compare the performance of the proposed approach against Bera et al. It is only the SOTA approach that allows for the desired $\boldsymbol{\tau}$-ratio fairness in a restrictive manner. Bera et al. reduces the degree of freedom using δ parameter that controls the lower and upper bound on number of data points needed

in each cluster belonging to a protected group. Experimentally δ can take values only in terms of dataset proportion r_ℓ for protected group $\ell \in [m]$, i.e. with lower bound as $r_\ell(1-\delta)$ and upper bound as $\frac{r_\ell}{(1-\delta)}$. Further, δ needs to be the same across all the protected groups, making it infeasible to achieve different lower bounds for each protected group. Thus Bera et al. cannot be used to have any general fairness constraints for each protected group and can act as a baseline only for certain τ_ℓ values. In Table 3.2, we present results for the $\boldsymbol{\tau}$ corresponding to δ=0.2, 0.8.

Observation: Our algorithms can achieve any generalized $\boldsymbol{\tau}$ vectors like $[0.25, 0.12]$. Such vectors make more sense in real-world applications, like requiring at least 25% male and 12% female data points in each cluster. The objective cost obtained by FRAC and FRAC$_{OE}$ is comparable to Bera et al., but the work by Bera et al. is extendible to multiple multi-valued protected groups.

Dataset	τ**- vector**	**FRAC**	**FRAC**$_{OE}$	**Bera et al.**	
		Objective Cost	**Objective Cost**	δ **Value**	**Objective Cost**
Adult	<0.133, 0.066 >	9804.65 ± 221.05	9616.51 ± 111.49	0.8	9515.30 ± 19.94
	<0.535, 0.264 >	10010.39 ± 211.27	10011.78 ± 239.73	0.2	9788.73 ± 23.32
	<0.25, 0.12 >	9870.93 ± 261.24	9714.06 ± 157.45	*Cannot be computed*	
Bank	<0.121, 0.056, 0.022 >	9210.38 ± 640.76	9043.51 ± 461.23	0.2	9588.30 ± 48.82
	<0.485, 0.225, 0.089 >	10982.63 ± 1228.28	11317.61 ± 1310.32	0.8	8472.65 ± 37.30
	<0.25, 0.10, 0.04 >	9548.68 ± 540.86	9465.35 ± 476.88	*Cannot be computed*	

Table 3.2: k-means objective cost for $\boldsymbol{\tau}$-ratio for adult and bank dataset for k=10 clusters.

3.7.4 Run-time Analysis

Finally, we compare the run-time of the different approaches for the k(=10)-means clustering versions on the Adult dataset. The average run-time over 10 different runs is reported in Figure 3.11 (b).

Observations:

- Run-time of FRAC is significantly better than the fair `SOTA` approaches.
- Ziko et al. (tuned) runtime is quite high due to hyperparameter tuning.
- Ziko et al. (untuned) is comparable to vanilla clustering but at deterioration in fairness (seen in previous sections).
- FRAC$_{OE}$ has a marginal difference from vanilla runtime as it applies a single round of fair assignment.
- Bera et al. being LP formulation has higher complexity and requires double the time of FRAC.

Motivated by Kriegel et al. [185], we further study the runtime behaviour across varying numbers of data points and varying numbers of clusters. For the scalability study, we

perform the analysis using Census-II as it is the largest dataset. We use the same hyper-parameter value (λ=500000) for Ziko et al. in this study.

Runtime comparison with number of cluster(k)

In this study, we conduct an experiment to find the variation in runtime as the number of clusters k varies from 2 to 40. We observe the results for $2, 5, 10, 15, 20, 30$ and 40. From the results summarized in Figure 3.12, we can observe that Bera et al. has a significantly high execution time. Thus, we limit the results up to $k(=5, 10)$-clustering. As pointed out in the previous section Bera et al., LP fails to converge for k=2.

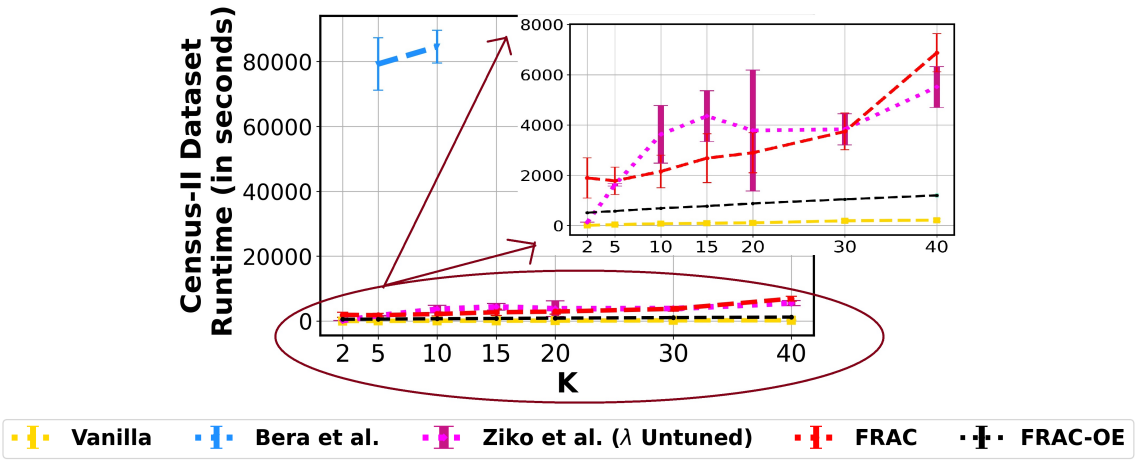


Figure 3.12: The line plot shows the variation of runtime over a varying number of clusters (k) for k-means setting on the complete dataset. The hyper-parameter λ=500000 is taken the same as that is reported in Ziko et al. paper for the Census-II dataset due to expensive computational requirements. For similar reasons, Bera et al. results for Census-II are evaluated for k=5 and k=10. For better visualization, the results are zoomed out for approaches other than Bera et al. (Best viewed in color).

Observations:

- FRAC$_{OE}$ has runtime close to vanilla clustering.
- Ziko et al., even in untuned version has runtime close to FRAC. Tuning will result in a significant increase in overall runtime.
- Bera et al. has significantly higher runtime.

Runtime comparison across varying data set size

We study the scalability of different approaches to increase in the data set size for k=10. For Bera et al., plots in Figure 3.13 reveal that the run time significantly increases even with $500,000$ data points in the data set. So, we limit the study to this size.

Observations:

- Ziko et al. (untuned) runtime is close to vanilla clustering. However, the gap increases after a certain dataset size.

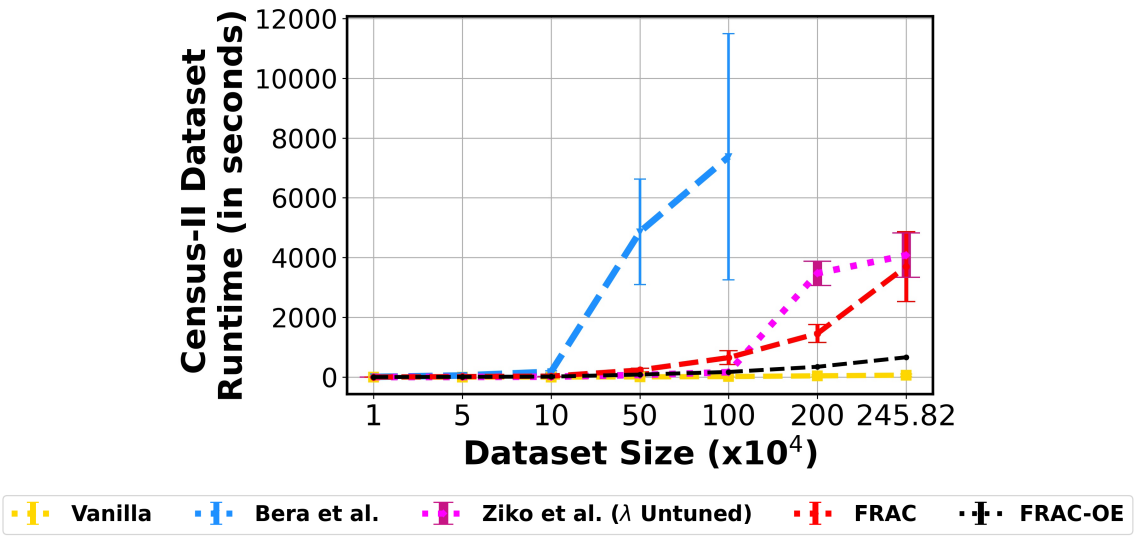


Figure 3.13: The line plot shows the variation of runtime over varying dataset size (up to complete dataset size of 245.82×10^4) for k=10-means setting. The hyper-parameter λ=500000 is taken the same as that is reported in Ziko et al. paper for the Census-II dataset due to expensive computational requirements. For similar reasons, Bera et al. results for Census-II are evaluated for dataset sizes of $10,000$, $50,000$, and $100,000$. (Best viewed in color).

- FRAC$_{OE}$ follows a trend slightly close to vanilla clustering and does not deteriorate with size, showing its efficiency.
- FRAC has a run time larger than vanilla clustering but is comparable to untuned Ziko et al..
- Tuning Ziko et al. will result in additional overhead.

3.8 Experimental Validation of Relationships between Fairness Levels and their Notions

This section validates the established theoretical underpinnings between different group fairness notions in Section 3.3. This is followed by the relationship between the group and individual fairness levels.

3.8.1 Relationship between Group Fairness Notions

Now, we empirically examine the relationship between different group fairness notions on *adult* and *bank* datasets. We fix k=10 and consider k-means clustering.
Many existing algorithms achieving group fairness are either limited to binary-protected groups [42, 11], require extensive hyper-parameter tuning [10, 93], or have high computational complexities [12, 13, 92, 95, 96]. So, we use our proposed polynomial-time algorithm FRAC$_{OE}$ [44], which supports multi-valued protected group for the study. The FRAC$_{OE}$ takes an input value τ_a for each protected group value $a \in [m]$ (see

Definition 3.5). We now ask the question - *Does satisfying the* τ-ratio *fairness notion helps to achieve other group fairness notions?* To answer this, we vary τ_a from 0 to $1/k$ (maximum achievable value) and study the induced levels of other group fairness notions. For simplicity, we fix τ_a to be a constant for all possible group values. The results on both datasets are averaged on five independent runs and plotted in Figure 3.14 along with standard deviation. From the plots, it is clear that the clusters satisfying τ-ratio fairness also satisfy high BALANCE guarantees, maintain lesser restricted dominance, and promote minority protection. The highest value of τ_a leads to maximally balanced clusters (*dataset ratio*).

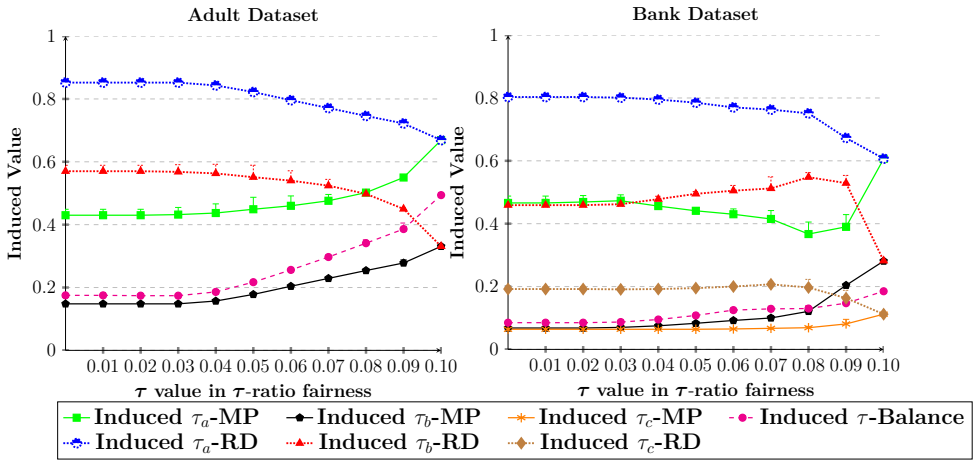


Figure 3.14: Induced group fairness values on k(=10)-means. (Best viewed in color)

We execute the linear program (LP) by Bera et al. [12] formulated to satisfy MP and RD and observe the τ-ratio fairness level. A remarkable observation is that satisfying MP and RD can lead to a degenerate value of τ-ratio fairness. This will happen when one cluster has very few data points from each group (maybe 1), and other clusters contain more data points, resulting in highly skewed clusters. Skewed clusters can be problematic in some cases. For example, in problems like direct marketing campaign [181], group fair clustering can be used to segment customers. Highly skewed clusters might not be profitable to invest in for customized solutions. However, using τ-ratio fairness guarantees a minimum number of data points (customers) from each group, i.e., a minimum cluster size while maintaining Balance (induced). This shows that τ-ratio is a stronger notion.

3.8.2 Relationship between Individual Fair Notions

We next show the connection between individual fairness notions. The results directly follow from definitions.

Result 3.22 (1)**.** *If* $x_i \in X$ *is* α*-PP then* x_i *is also* α*-*AG*.*

Proof. Since the average value of a set is always larger than the minimum set value, we have:

$$\min_{x'_i \in S(x_i)} d(x'_i, \phi(x'_i)) \leq \frac{\sum_{x'_i \in S(x)} d(x'_i, \phi(x'_i))}{|S(x)|}.$$

Therefore, if x_i is α-PP fair, then x_i is also α-AG fair. □

3.8.3 Relationship between Group and Individual Fairness Level

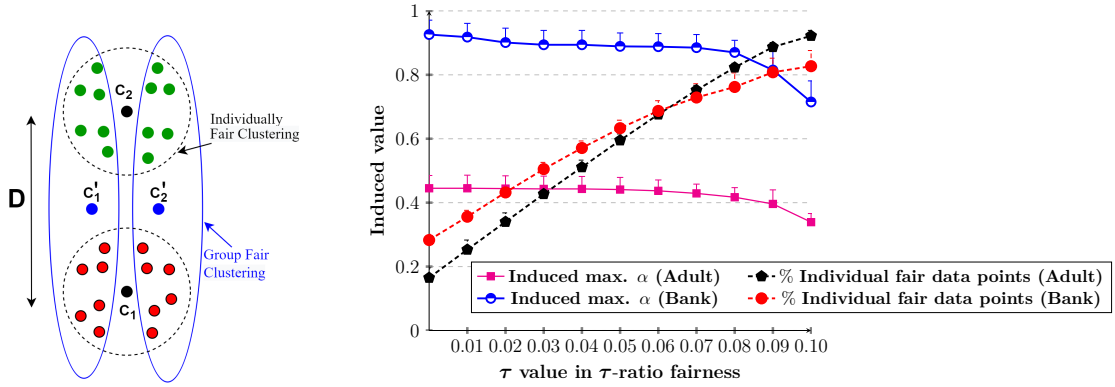
The two fairness levels arose independently in fair clustering literature. Nonetheless, many real-world applications demand satisfying both group and individual fairness. In direct marketing, the corporate house's diversity policy necessitates group fairness. But at the same time, customers might feel discontented if people in their similarity set belong to a different cluster than their own (hence offering different benefits). Thus, there is a need to study the relationship between the two levels.

Recent attempts [64, 65] explore this direction and propose instances that show the conflicting nature of both the fairness levels, i.e., satisfying one might adversely affect the other. To understand this, consider a dataset with data points split across two far-apart clusters, with each cluster containing data points from one protected group (as illustrated in Figure 3.15(a)). Group fair clustering will try to place the cluster centers in between the two clusters. On the contrary, the original cluster centers will also serve as optimal individual fair centers when the individual fairness notions depend on distance-based similarity. Thus, showing both fairness as conflicting problems.

We experimentally study the induced individual fairness effect by trying to satisfy group fairness. The reverse trend follows without loss of generality. We use k-means version of FRAC$_{OE}$ algorithm for k=10. We report the maximum deviation value (i.e., α in α-FR) and the fraction of data points satisfying the α-FR (Definition 2.6) with α=1 to the total number of data points. Figure 3.15 (b) shows the mean and standard deviation over five runs. The plots show that both fairness levels are not strictly in conflict when evaluated on real-world datasets. They both induce certain levels of fairness in the clusters. For both datasets, the number of data points having strict individual fairness of having a center within a given radius increases significantly with an increase in $\boldsymbol{\tau}$. Further, this shows that very few data points have large violations, i.e., the maximum α value reported limits to a small set of data points. We study the direction of approximating multiple fairness levels in next chapter in detail.

3.9 Conclusion

The chapter proposed a novel $\boldsymbol{\tau}$-ratio fairness notion. The new notion is a stricter variation of the existing group fairness notion and admits an efficient round-robin algorithm to the corresponding fair assignment problem. We also showed that our proposed algorithm, FRAC$_{OE}$, (i) achieves $2(\beta+2)$-approximate solution up to three clusters, and (ii) achieves $2^{k-1}(\beta+2)$-approximate guarantees to general k with τ=$1/k$ for all protected group values. Current proof techniques for $k \leq 3$ require intricate case analysis, which becomes intractable for larger k. However, our experiments show that FRAC$_{OE}$ and FRAC outperform `SOTA` approaches in objective cost and fairness measures even for k >3. We also prove the cost approximation for the general τ vector and show convergence analysis



(a) (b)

Figure 3.15: (a) Example illustrating conflicting group and individual fair clustering. Here C_1, C_2 are individual fair centers separated by large distance D, and C'_1, C'_2 are group fair centers. (b) Induced α-FR individual fairness values on $k(=10)$-means.

for FRAC_{OE}. Other than these, the chapter also experimentally validates the relationship between $\boldsymbol{\tau}$-ratio and existing group fairness notions. The results on real-world datasets show that satisfying $\boldsymbol{\tau}$-ratio also ensures Balance, MP and RD properties, showing the strictness of the proposed notion. We also show that though individual and group level fairness seems complementary. However, our results on real-world datasets show that using FRAC_{OE} with $\boldsymbol{\tau}$-ratio fairness induces a certain level of individual fairness. It motivates us to carefully investigate multiple levels of fairness, particularly in real-world settings. We will study this direction in the next chapter on facility location problems. An immediate future direction is to analytically prove $2(\beta + 2)$-approximation guarantee for general k. Other interesting future directions include extending the current work to multi-valued multiple protected groups like the one proposed by Bera et al. [12] or achieving group fairness in strategic settings [186] or under the presence of noisy feedback for protected group values [187].

Chapter 4

Balancing Fairness and Efficiency via Novel Welfare Perspective

Abstract

The Facility Location Problem (FLP) identifies the most suitable facility locations and assigns agents to different facilities. Recent studies shows evidence of biases in facility location problem based on agents' group memberships, such as gender or income, resulting in adverse consequences. Group fairness ensures that each facility is assigned a minimum fraction of agents from every group. In order to foster a more individually equitable allocation of agents to facilities, we adopt a novel formulation motivated by Nash social welfare instead of considering standard utilitarian or egalitarian approaches. We propose an efficient and scalable algorithm called `FAIRLOC` *(Fair Algorithm for Facility Location) that minimizes the product of distances (or costs) of agents to assigned facilities while obeying group fairness constraints. We theoretically provide approximation bounds on cost with respect to optimal fair allocation and show that* `FAIRLOC` *achieves a quadratic approximation in the product-based objective function. With the help of extensive experimentation of real-world U.S. geography datasets using Open Source Routing Machine (OSRM) roadmaps, we show that* `FAIRLOC` *achieves significantly lower costs and better group and individual fairness metrics than state-of-the-art methods.*

4.1 Introduction

Motivated by the success of achieving group fairness in clustering, we look into a real-world application of the facility location problem in this chapter. The problem addresses the practical challenge of determining the most suitable locations for facilities (such as shops, offices, etc.) to serve the needs of agents (consumers/workers) [188]. Closely related to clustering, it requires finding facility opening locations (centers) that minimize the travel time (or cost) for agents (data points) to access the facility. Prior works have even attempted solving facility location problems using clustering [88]. We build upon this direction and look into the facility location problem, particularly with agents preferring a facility closer to them. To understand the need for fairness constraints while identifying

The paper from this chapter is under review.

optimal locations in FLP, consider the following scenario:- Suppose the federal government wants to set up k public shelters for refugees across their states. The agents (refugees) currently are spread around in different locations across these states, and the primary objective of government planners is to strategically locate the shelters to reduce agents' distance to assigned facilities. Additionally, to prevent discontent and negative feelings among state agencies and nearby residents, government planners must ensure that no facility becomes dominated by agents of a specific type (say, based on protected groups such as race or ethnicity). Recent studies have revealed the existence of favouritism toward certain groups of agents based on sensitive attributes (or protected groups) such as gender, age, income level, race, etc. [189, 190, 191, 192]. For example, member states in the European Union (EU) have pledged to host a minimum number of relocations to ensure solidarity and a fair share of responsibility. It has resulted in strengthened partnerships and efficient asylum systems. To further improve the effectiveness of such schemes, Efthymiou [193] argue that refugees can be better protected and integrated if the EU focuses on constraining the number of refugees from specific protection, say religion or ethnicity, rather than just extensively maximizing quota numbers. The main argument to support the need for balanced representation (which they call robust conception) is as follows: First, in the case of breading agents of a single type, due to geopolitical and socioeconomic factors, is short-sighted. Such states might shift their commitment once the associated rewards get altered. Secondly, a more balanced representation of refugees at each shelter reduces the chances of minorities (across the state) feeling biased and misled. Instead, it fosters trust and long-term commitment and embarks a healthy sensation in the minds of refugees.

A need for balanced assignments has also been observed in the supermarket chains [194, 195, 196, 197], vaccine distribution sites, dialysis centers, and emergency rooms [198, 199, 200]. To tackle the need for the desired level of coverage, this chapter considers *group fairness* notion, which ensures a minimum representation of agents from each protected group value (say male and female in gender) at every facility. Additionally, most facility location problem research has emphasized objectives such as utilitarian (sum of distances of agents to assigned facilities) or egalitarian (minimizing the maximum distance). This work proposes a novel adaptation of the Nash social welfare to the facility location problem. Nash social welfare is a well-studied notion in various fair resource allocation and fair division literature [201, 202] and involves the product of the costs. Nash social welfare provides a balance between utilitarian and egalitarian objectives and prefers a more balanced assignment profile (in terms of distances to facilities) of all agents than a skewed one. In particular, the **key contributions** of our Nash social welfare modelling are as follows:

- Proposing a first-of-its-kind application of modeling Nash social welfare to facility location problem to target a more equitable allocation of agents to facilities by minimizing agents' distance under group fairness constraints.
- Proposing an efficient algorithm, `FAIRLOC` that solves facility location problem in

h-dimensional space and maximizes Nash social welfare subject to group fairness constraints. `FAIRLOC` does not make any assumptions on facility locations and allows the use of an explicit facility opening locations set.

- Theoretically, `FAIRLOC` achieves a bounded approximation on cost guarantees to optimal fair allocation.

- `FAIRLOC` performs significantly better on individual and group fairness metrics with lower costs as compared to state-of-the-art methods on near real-world testing on the United States census dataset with road maps providing the actual car road distances between agents and facilities.

4.2 Related Work

4.2.1 Facility Location Problem

Facility location problems have seen continuous development for the past decades, and for more details, readers can refer to [203, 204]. Despite being an NP-Hard problem [205], the solutions to facility location problem include approximation algorithms [206], integer (or mixed) integer programming [207], greedy algorithms [208], clustering [88], and others. The past literature has primarily tackled the facility location problem by optimizing either the egalitarian (the maximum distance of any agent) or the utilitarian objective (the sum of distances of all agents) [209, 210, 211].

4.2.2 Fairness in Facility Location Problem

Marsh and Schilling [212] reviews existing metrics for addressing group fairness and concludes that there is no universal consensus on a single metric, and the choice of metric depends on the application and problem. In this chapter, we use our proposed $\boldsymbol{\tau}$-ratio metric for clustering, which is a stricter notion of group fairness. Also, the choice is driven by its success in achieving a polynomial time algorithm in the clustering setup. Prior works, such as Li et al. [213] and Zhou et al. [214], explore group fair facility location problem in one-dimensional space and are limited to settings where only one facility needs to be placed (i.e., 1-facility problem). Jung et al. [88] introduced the notion of fairness based on the density of agents in the space. The authors proved that achieving this notion of fairness while maintaining the standard utilitarian objective is NP-hard, and they proposed a 2-approximation algorithm to satisfy this fairness objective. The concept later became widely known as individual fairness [97, 215, 91]. One of the closest works using Nash social welfare in facility location problem is by Lam et al. [202], which is limited to agents in a one-dimensional space and one facility. We employ a Nash social welfare-based formulation to achieve standard facility location problem goals while opening multiple (k) facilities and obeying the notion of group fairness.

4.3 Preliminaries

In this section, we mainly discuss the notations and definitions that will help better understand the proposed algorithms and its theoretical guarantees.

4.3.1 The Model and Notation

Let $X \subseteq \mathbb{R}^h$ be a set of n agents located in any h-dimensional space. Each agent is associated with a single protected group, such as gender or income level, which can take a value from the set of m values denoted by $[m]$. The mapping function $\rho : X \to [m]$ provides the protected group of each agent. Let n_ℓ, X_ℓ be the number and set of agents from group ℓ. Also, we denote the set of facilities to be opened as $L \subseteq F$ of k facilities (say, hospitals). The primary goal is to find a set L and design an assignment function $\phi : X \to L$. We capture an agent's preference for different facilities as closeness between the agent's location ($x_i \in X$) to their assigned facility ($f \in F$) and is measured using distance metric $d : X \times F \to \mathbb{R}^+ \cup \{0\}$.

4.3.2 Fairness in Facility Location Problem

Group Fairness To tackle group fairness, we chose $\boldsymbol{\tau}$-ratio [44], which we now redefine for the sake of completeness and better readability.

Definition 4.0 ($\boldsymbol{\tau}$-ratio Fairness)

An assignment function ϕ obeys *$\boldsymbol{\tau}$-ratio fairness* if for a given vector $\boldsymbol{\tau} = \{\tau_1, \tau_2, \ldots, \tau_\ell, \ldots, \tau_m\}$ and $\forall f \in L$ we have:

$$\sum_{x_i \in X_\ell} \mathbb{I}(\phi(x_i) = f) \geq \tau_\ell n_\ell \ \forall \ell \ \in [m]$$

Individual Fairness Since optimizing Nash social welfare prefers more equitable assignments, we would also like to see how `FAIRLOC` performs on individual fairness metrics [88]. We use the α-`FR` notion and redefine it as follows:

Definition 4.1 (Individually Fair Radius (r))

Given X, ϕ with metric d, for every agent $x_i \in X$, we define fair radius $r(x_i)$ as the minimum distance around x_i such that $|\mathcal{B}(x_i, r(x_i))| \geq \lceil n/k \rceil$ where $\mathcal{B}(x_i, r(x_i)) = \{x_j \in X : d(x_i, x_j) \leq \alpha \cdot r(x_i)\}$.

4.3.3 Welfare Functions

Definition 4.2 (Utilitarian Objective)

This objective minimizes the total distance (or cost) of all agents, i.e., $\min_{L,\phi} \sum_{x_i \in X} \Big(d(x_i, \phi(x_i))\Big)$.

Definition 4.3 (Egalitarian Objective)

This objective minimizes the maximum distance of an agent, i.e., $\min_{L,\phi}(\max_{x_i \in X} \Big(d(x_i, \phi(x_i))\Big))$.

Motivated by the success of Nash Social Welfare in social choice theory, we adapt the welfare function to facility location problems to minimize the distance of each agent to the assigned facility.

Definition 4.4 (Nash Social Welfare)

This objective minimizes the product (or geometric mean) of distances of agents, i.e.,

$$\min_{L,\phi} NW(X, F, \phi) = \min_{L,\phi} \left(\prod_{x_i \in X} d(x_i, \phi(x_i)) \right)^{1/n} \tag{4.1}$$

Nash objective results in more equitable and evenly distributed allocations i.e. it does not favour allocations in which an increase in an agent's cost is significantly more than a decrease in any other agent's cost. In contrast, utilitarianism focuses on the sum of distances and allows a substantial reduction in an agent's cost that can be compensated by increased costs incurred for others. Thus, Nash's welfare is more individually fair than utilitarian. In the next section, we discuss in detail the proposed algorithm, which we call `FAIRLOC`- Fair Algorithm for Facility Location that optimizes Nash welfare cost while obeying group fairness ($\boldsymbol{\tau}$-ratio fairness) constraints.

4.3.4 Proposed Mathematical Model

Our **optimization problem** with $z_{x,f}$ and y_f as decision variables is given as:

$$\min_{z_{x,f}, y_f} \left(\prod_{x \in X} (\sum_{f \in F} d(x, f) \cdot z_{x,f}) \right) \tag{4.2}$$

$$\text{s.t.} \sum_{f \in F} z_{x,f} = 1, \quad \forall x \in X \tag{4.3}$$

$$\sum_{f \in F} y_f = k \tag{4.4}$$

$$z_{x,f} \leq y_f, \quad \forall f \in F, \forall x \in X \tag{4.5}$$

$$\tau_\ell n_\ell y_f \leq \sum_{x \in X_\ell} z_{x,f}, \ \ \forall \ell \in [m] \ \ \forall f \in F \tag{4.6}$$

Equation 4.2 corresponds to the objective function. The constraints in Equations 4.3 and 4.4 ensure that each agent is assigned to exactly one facility and exactly k facilities are opened, respectively. While the constraints in Equations 4.5 and 4.6 ensure that a facility is opened if an agent is assigned to it and group fairness constraint respectively.

4.4 Proposed Algorithm: FAIRLOC

Facility location and clustering are hard problems, so approximation algorithms are proposed in the literature [216, 42]. Our algorithm is motivated by FRAC$_{OE}$ – designed to minimize utilitarian objective subject to $\boldsymbol{\tau}-$fairness constraint. FRAC$_{OE}$ for clustering works by allocating agents according to their distance from each protected group in a round-robin fashion to the centers obtained by traditional clustering algorithms.

Difference between FAIRLOC and FRAC$_{OE}$. FAIRLOC uses the same approach as FRAC$_{OE}$ but replaces the traditional clustering algorithm with the algorithm that focuses on optimizing Nash social welfare. It must be noted that while the FAIRLOC looks similar to that of FRAC$_{OE}$, the theoretical guarantees of FRAC$_{OE}$ are no longer applicable due to the change in the objective function. Further, while FRAC$_{OE}$ makes an assumption that available facility locations set F coincide with agent location set X, FAIRLOC does not use this assumption. The complete pseudo-code for FAIRLOC method is provided in the Algorithm 4.

FAIRLOC computes the initial facility locations L (**Initial_Nash_Locations**) by starting with random initialization of k facilities and assigning an agent to the closest facility. Once all the agents are allocated to the initial facilities, let X_f denote the set of data points assigned to facility f. The new location corresponding to the location f is updated as follows (**Update_Nash_Locations**): $f^{'} = \text{argmin}_{f' \in F} \left(\prod_{x_i \in X_f} d\left(x_i, f^{'}\right) \right)$ resulting in formation of set $L^{'}$. These steps are repeated until convergence or maximum iterations T. INITIAL_NASH_LOCATIONS finally returns a converged set of facility locations denoted by L.

Note that when $F = X$, minimizing the traditional utilitarian function for the update rule is comparatively easier, as the mean of the data points minimizes the sum of the distances [217]. However, when $F \neq X$ and the objective is to minimize the product of distances, solving the derivatives of the objective does not lead to a closed-form solution. When distance metric is $d(x_i, f) = ||\log(x_i) - \log(f)||^2$ [218], then it can be shown that geometric mean minimizes the product of distances when $F = X$. One way to convert $F = X$ setting to $F \neq X$ setting is by mapping each facility obtained from the first setting to the nearest setting in set F. However, it may not lead to optimal values of respective objective function values. In the current work, for theoretical guarantees, we use standard p-norm metric such as Euclidean distances, which are known to hold desirable properties such as triangular inequalities, symmetric, and positiveness. However, FAIRLOC will work for any distance metric. FAIRLOC update the best possible location within each set of assignments for a particular facility by trivially checking among all possible locations as

Algorithm 4: `FAIRLOC`

Input: set of agent location X, set of possible facility opening locations F, number of facilities to open k, group fairness requirement $\boldsymbol{\tau}$, protected group function ρ, distance function d, maximum iterations T

Output: assignment function ϕ, facility opening location set L

Initialize $\phi[x_i] \leftarrow \Phi \quad \forall x_i \in X$
$L, \hat{\phi} \leftarrow$ INITIAL_NASH_LOCATIONS(X, T, F)
`//Reassignment to satisfy` $\boldsymbol{\tau}$`-fairness.`
for $\ell \in [m]$ **do**
  $X_\ell \leftarrow \{x_i : x_i \in X \text{ and } \rho(x_i) = \ell\}$; $\quad n_\ell \leftarrow |X_\ell|$
  for $r \leftarrow 1$ **to** $\tau_\ell n_\ell$ **do**
    for $f \in L$ **do**
      $x_i \leftarrow \text{argmin}_{x_i \in X_\ell:\ \phi[x_i]=\Phi}\ d(x_i, f)$
      $\phi[x_i] = f$
    end
    $r = r + 1$
  end
end
for $x_i \in X$ **do**
  if $\phi[x_i] = \Phi$ **then**
    $\phi[x_i] \leftarrow \hat{\phi}[x_i]$
  end
end
$L =$ UPDATE_NASH_LOCATIONS$(X,\ \phi,\ \text{F})$
return ϕ, L

evident in line 4 of Algorithm 6. Note that such a brute search will not be computationally expensive as in most real-world settings, the number of facilities to open k and the size of explicit possible locations (F) is quite less compared to the size of X.

After computing the facility locations (L) and initial assignment ($\hat{\phi}$), we convert the fair facility location problem to a fair facility assignment problem. The goal of the assignment problem is to redistribute the agents to facilities in L set to maintain $\boldsymbol{\tau}$-ratio fairness. We theoretically show through the following lemma that converting the fair facility location problem to a fair facility assignment problem leads to a quadratic approximation when the objective is a product of distances (or two approximation of the logarithmic sum of distances).

Lemma 4.1. *Let $\mathcal{I}$ be an instance of a fair facility location problem and $\mathcal{T}$ an instance of $\boldsymbol{\tau}$-ratio fair assignment problem after applying an β-approximate solution to the vanilla (unfair) facility location problem, then $\mathcal{OPT}_{assign}(\mathcal{T}) \leq ((2^n + 1)\beta)^{1/n}(\mathcal{OPT}_{FLP}(\mathcal{I}))^2$. Here $\mathcal{OPT}_{FLP}(\cdot)$ denotes the Nash cost for the fair facility problem and $\mathcal{OPT}_{assign}(\cdot)$ for the fair assignment problem on the input instance.*

Proof. Let L be the facility locations obtained by running a vanilla facility algorithm on instance $\mathcal{I}$. The proof of the Lemma depends on the existence of an assignment ϕ satisfying $\boldsymbol{\tau}$-ratio fairness such that $NW(X, F, \phi) \leq ((2^n+1)\cdot\beta)^{1/n} \cdot (\mathcal{OPT}_{FLP}(\mathcal{I}))^2$. Then it follows

as $\mathcal{OPT}_{assign}(\mathcal{T}) \leq NW(X, F, \phi) \leq ((2^n + 1) \cdot \beta)^{1/n} \cdot (\mathcal{OPT}_{FLP}(\mathcal{I}))^2$
To this, let (F^*, ϕ^*) denote the optimal solution to $\mathcal{I}$. Define ϕ as follows: for every $f^* \in F^*$, let $nrst(f^*) = \text{argmin}_{f \in F}\, d(f, f^*)$ be the nearest center to f^*. Then, for every $x_i \in X$, define $\phi(x_i) = nrst(\phi^*(x_i))$. Then we have the following two claims:

Claim 4.2. *ϕ satisfies $\boldsymbol{\tau}$-ratio fairness.*

Proof. The proof is same as the Claim 3.9 from previous chapter. □

Claim 4.3. $NW(X, F, \phi) \leq ((2^n + 1) \cdot \beta)^{1/n} \cdot (\mathcal{OPT}_{FLP}(\mathcal{I}))^2$

Proof. For agent $x_i \in X$, we have $\hat{f} = \hat{\phi}(x_i)$ as facility location after applying vanilla (unfair) facility location solution, f as facility location after using `FAIRLOC`'s (including fairness procedure), and f^* be the facility location using fair optimal facility location solution. Then we have,

$$d(x_i, f) = d(x_i, nrst(f^*)) \leq d(x_i, f^*) + d(f^*, nrst(f^*)) \leq 2 \cdot d(x_i, f^*) + d(x_i, \hat{f}\ ) \quad (4.7)$$

The above equations use triangular inequalities and the definition of $nrst(\cdot)$. We now look into the bound on the complete set of agent locations $x \in X$. Therefore, we have,

$$\left(\prod_{i=1}^{n} d(x_i, f)\right)^{1/n} \leq \left(\prod_{i=1}^{n} \left(2 \cdot d(x_i, f^*) + d(x_i, \hat{f}\ )\right)\right)^{1/n} \qquad \text{(using Equation 4.7)}$$

Now for ease of reading consider $a_i = 2 \cdot d(x_i, f^*)$ and $b_i = d(x_i, \hat{f}\ )$. So one needs to expand the term of the form $\prod_{i=1}^{n}(2a_i + b_i)$. We bound this as follows:

$$\prod_{i=1}^{n}(2a_i + b_i) = \prod_{i=1}^{n} 2 \cdot a_i + 2 \cdot a_1 \cdot \prod_{j \neq 1} b_j + 2 \cdot a_2 \cdot \prod_{j \neq 2} b_j + \ldots + 2^2 \cdot a_1 \cdot a_2 \cdot \prod_{j \neq 1,2} b_j + \cdots + \prod_{i=1}^{n} b_i \quad (4.8)$$

Now, we know that $\prod_{i=1}^{n} 2 \cdot a_i \leq 2^n \cdot (\mathcal{OPT}_{FLP}(\mathcal{I}))^n$ and $\prod_{i=1}^{n} b_i \leq \beta \cdot (\mathcal{OPT}_{FLP}(\mathcal{I}))^n$ where β is the approximation factor of vanilla (unfair) facility location problem. Using these two results, we provide a quite loose upper bound on the other 2^{n-1} terms such as for $a_1 \cdot \prod_{j \neq 1} b_j \leq \beta(\mathcal{OPT}_{FLP}(\mathcal{I}))^{2n}$. Similarly, applying for all inner 2^{n-1} terms involved in the expansion, we get as below:

$$\begin{aligned}\prod_{i=1}^{n}(2a_i + b_i) &\leq 2^n \cdot (\mathcal{OPT}_{FLP}(\mathcal{I}))^n + 2 \cdot \beta \cdot (\mathcal{OPT}_{FLP}(\mathcal{I}))^{2n} + \ldots + 2^2 \cdot \beta \cdot (\mathcal{OPT}_{FLP}(\mathcal{I}))^{2n} \\ &\quad + \ldots + \beta \cdot (\mathcal{OPT}_{FLP}(\mathcal{I}))^n \qquad (4.9) \\ &\leq 2^n \cdot (\mathcal{OPT}_{FLP}(\mathcal{I}))^n + 2^{n-1} \cdot 2 \cdot \beta \cdot (\mathcal{OPT}_{FLP}(\mathcal{I}))^{2n} + \beta \cdot (\mathcal{OPT}_{FLP}(\mathcal{I}))^n \qquad (4.10)\end{aligned}$$

Algorithm 5: INITIAL_NASH_LOCATIONS

Input: set of agent location X, set of possible facility opening locations F, maximum iteration T

Output: converged facility opening location set L

$L \leftarrow$ Choose k random agent location from F
while $t < T$ *or* $L \neq L_{prev}$ **do**
  for $x_i \in X$ **do**
    $\phi(x_i) = \text{argmin}_{f \in L}\, d(x_i, f)$
  end
  $L^{'} \leftarrow$ UPDATE_NASH_LOCATIONS(X, ϕ, F)
  Set $L_{prev} = L$; $\quad L = L^{'}$; $\quad t = t + 1$;
end
return L

Algorithm 6: UPDATE_NASH_LOCATIONS

Input: set of agent location X, assignment function ϕ, set of possible facility opening locations F

Output: updated facility opening location set L'

Initialize $L^{'} \leftarrow \Phi$;
for $f \in L$ **do**
  $X_f = \{x_i : x_i \in X \text{ and } \phi(x_i) = f\}$
  $\ell^{'} = \text{argmin}_{\ell^{'} \in F}(\prod_{x_i \in X_f} d(x_i, f^{'}))$ $L^{'} \leftarrow L^{'} \cup \{\ell^{'}\}$
end
return $L^{'}$

$$= 2^n \cdot (\mathcal{OPT}_{FLP}(\mathcal{I}))^n + 2^n \cdot \beta \cdot (\mathcal{OPT}_{FLP}(\mathcal{I}))^{2n} + \beta \cdot (\mathcal{OPT}_{FLP}(\mathcal{I}))^n \tag{4.11}$$

$$\leq 2^n \cdot (\mathcal{OPT}_{FLP}(\mathcal{I}))^{2n} + 2^n \cdot \beta \cdot (\mathcal{OPT}_{FLP}(\mathcal{I}))^{2n} + \beta \cdot (\mathcal{OPT}_{FLP}(\mathcal{I}))^{2n} \tag{4.12}$$

$$\leq ((2^n + 1) \cdot \beta) \cdot (\mathcal{OPT}_{FLP}(\mathcal{I}))^{2n} \tag{4.13}$$

$$\implies \left(\prod_{i=1}^{n} d(x, f)\right)^{1/n} \leq \left((2^n + 1) \cdot \beta\right)^{1/n} \cdot \left(\mathcal{OPT}_{FLP}(\mathcal{I})\right)^2 \tag{4.14}$$

$$\implies NW(X, F, \phi) \leq \left((2^n + 1) \cdot \beta\right)^{1/n} \cdot \left(\mathcal{OPT}_{FLP}(\mathcal{I})\right)^2 \tag{4.15}$$

□

□

To efficiently solve the assignment problem, we fix a random ordering (experimentally performance invariant, Section 4.6.6) over the facilities. Then, `FAIRLOC` allocates the available agent with the lowest distance to each facility in a round-robin fashion for $\tau_\ell n_\ell,\ \forall \ell \in [m]$ rounds to each facility. Here n_ℓ is the number of agents of type ℓ in the dataset. This ensures group fairness guarantee by distributing at least a τ_ℓ fraction of agents at each facility. The remaining agents are allocated by assigning the agent

to the location that minimizes its distance (or cost), i.e., ($\phi = \hat{\phi}$). Next, we provide approximation bounds on the FAIRLOC's assignment.

4.5 Theoretical Results

We now provide the theoretical guarantees of FAIRLOC with respect to $\boldsymbol{\tau}$-ratio fairness. We first provide guarantees for maximally balanced facilities, i.e. $\tau_\ell = 1/k \ \forall \ell \in [m]$ setting and later extend these to general $\boldsymbol{\tau}$ vector.

Theorem 4.4. *Let $k = 2$ and $\tau_\ell = \frac{1}{k}$ for all $\ell \in [m]$. An allocation returned by FAIRLOC guarantees $\boldsymbol{\tau}$-ratio fairness and satisfies $3^{1/4}\vartheta^{3/4}(\mathcal{OPT}_{assign})^2$-approximation guarantee to the product of distances with respect to an optimal fair assignment with ϑ being an instance-dependent multiplicative constant.*

Proof. **Correctness and Fairness:** Clear from the construction of the algorithm.
Proof of (approximate) Optimality: We will prove the approximation with respect to each value ℓ of protected group separately. Since n_ℓ is the number of agents corresponding to the value ℓ. We now show that $\texttt{FAIRLOC}(\mathcal{T}) \leq 3^{1/4}\vartheta^{3/4} \ (\mathcal{OPT}_{assign}(\mathcal{T}))^2$, where $\texttt{FAIRLOC}(\mathcal{T})$ and $\mathcal{OPT}_{assign}(\mathcal{T})$ denote the objective value of the solution returned by FAIRLOC and optimal assignment algorithm respectively on given instance $\mathcal{T} = (X, F)$. Let $\vartheta := 2\sup_{x,y\in X} d(x, y)$ be the diameter of the feature space. We begin with the following useful definition.

Definition 4.5 (Bad Assignments)

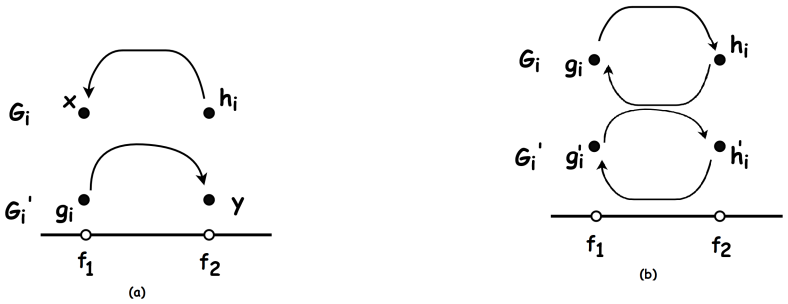
Let $\mathcal{C}_1$ and $\mathcal{C}_2$ represent the set of agents assigned to facilities f_1 and f_2 by optimal assignment algorithm[a]. The i^{th} round (i.e. assignments g_i to f_1 and h_i to f_2) of FAIRLOC is called

- 1-bad if exactly one of 1) $g_i \notin \mathcal{C}_1$ or 2) $h_i \notin \mathcal{C}_2$ is true, and
- 2-bad if both 1) and 2) above are true.

Furthermore, a round is called bad if it is either 1-bad or 2-bad and called good otherwise.

[a]Note that an optimal fair allocation need not be unique. Our result holds for any optimal fair allocation.

Let all incorrectly assigned agents in a bad round be called bad assignments. We use the following convention to distinguish between different bad assignments. If $g_i \notin \mathcal{C}_1$ holds we refer to it as type 1 bad assignment i.e. if agent g_i is currently assigned to $\mathcal{C}_1$ but should belong to optimal allocation $\mathcal{C}_2$. Similarly, if $h_i \notin \mathcal{C}_2$ holds it is a type 2 bad assignment i.e. h_i should belong to optimal allocation $\mathcal{C}_1$ but is currently assigned to f_2. Hence a 2-bad round results in 2 bad assignments one of each type i.e. $g_i \notin \mathcal{C}_1$ and $h_i \notin \mathcal{C}_2$. In summary, each 1-bad round can have either type 1 or type 2 bad assignment and each



(a) Two 1-bad round pairs (b) Two 2-bad round pairs

Figure 4.1: Different cases for $k = 2$. (a) Shows two 1-bad rounds with four assignments such that x, y are good assignments and allocated to the optimal facility by algorithm, whereas g_i and h_i are bad assignments with an arrow showing the direction to the optimal facility from the assigned center. (b) Shows four bad agents such that g_i, g_i' are assigned to f_1 but should belong to f_2 in optimal allocation (the arrow depicts the direction to optimal center). Similarly, h_i, h_i' should belong to f_1 in optimal allocation.

2-bad round will have two bad assignments each of type 1 and type 2. Finally, let B be the set of all bad rounds and A be the set of all bad assignments.

Definition 4.6 (Complementary Bad Pair)

A pair of agents $w, z \in A$ such that w is a bad assignment of type t and z is a bad assignment of type $|3-t|$ is called a complimentary bad pair if,
1) w and z are allocated in same round (i.e. in a 2-bad round) or
2) if they are allocated in i^{th} and j^{th} 1-bad rounds respectively with $i < j$, then z is the first bad assignment of type $(3-t)$ which has not been yet paired with a complementary assignment.

Lemma 4.5. *If n_ℓ is even, every bad assignment in the allocation returned by* `FAIRLOC` *has a complementary assignment. If n_ℓ is odd, at most one bad assignment will be left without a complementary assignment.*

Proof. The proof works on the same lines as proof of Lemma 3.12. □

We will bound the optimality of 1-bad rounds and 2-bad rounds separately.

Bounding 1-bad rounds: When n_ℓ is even, from Lemma 4.5, there are even number of 1-bad rounds; two for each complimentary bad pair. Let the 4 agents of corresponding two 1-bad rounds be $G_i : (x, h_i)$ and $G_i' : (g_i, y)$ as shown in Figure (4.1a). Note that $x \in \mathcal{C}_1$ and $y \in \mathcal{C}_2$ i.e. both are good assignments and $g_i \notin \mathcal{C}_1$, $h_i \notin \mathcal{C}_2$ are bad assignments. Now, consider an instance $\mathcal{T}_i = \{C, \{x, h_i, g_i, y\}\}$, then $\mathcal{OPT}_{assign}(\mathcal{T}_i) = \Big(d(x, f_1) \cdot d(h_i, f_1) \cdot d(g_i, f_2) \cdot d(y, f_2)\Big)^{1/n}$ considering nash motivated cost as product of distances. This implies $\Big(\mathcal{OPT}_{assign}(\mathcal{T}_i)\Big)^n = \Big(d(x, f_1) \cdot d(h_i, f_1) \cdot d(g_i, f_2) \cdot d(y, f_2)\Big)$ .We consider, without loss of generality, that the round G_i takes place before G_i' in the execution of `FAIRLOC`. The proof

is similar for the other case. First note that since FAIRLOC assigns h_i to facility 2 while both g_i and y were available, we have,

$$d(h_i, f_2) \leq d(g_i, f_2) \text{ and } d(h_i, f_2) \leq d(y, f_2) \tag{4.16}$$

So,

$$\begin{aligned}
\Big(\texttt{FAIRLOC}(\mathcal{T}_i)\Big)^n &= d(x, f_1) \cdot d(h_i, f_2) \cdot d(g_i, f_1) \cdot d(y, f_2) && (4.17)\\
&= d(x, f_1) \cdot d(h_i, f_2) \cdot d(y, f_2) \cdot d(g_i, f_1) && \text{(using Equation 4.16 and rearranging)}\\
&\leq d(x, f_1) \cdot d(h_i, f_2) \cdot d(y, f_2) \cdot \Big(d(g_i, f_2) + d(f_1, f_2)\Big) && (\because \text{triangle inequality})\\
&\leq d(x, f_1) \cdot d(h_i, f_2) \cdot d(y, f_2) \cdot \Big(d(g_i, f_2) + d(h_i, f_2) + d(h_i, f_1)\Big) && (4.18)\\
&\leq d(x, f_1) \cdot d(y, f_2) \cdot d(y, f_2) \cdot \Big(2 \cdot d(g_i, f_2) + d(h_i, f_1)\Big) && (4.19)\\
&\leq 2 \cdot \Big(\mathcal{OPT}_{assign}(\mathcal{T}_i)\Big)^{2n} + \Big(\mathcal{OPT}_{assign}(\mathcal{T}_i)\Big)^n \leq 3 \cdot \Big(\mathcal{OPT}_{assign}(\mathcal{T}_i)\Big)^{2n} && (4.20)
\end{aligned}$$

If n_ℓ is odd, then all the other rounds can be bounded using the above cases except one extra 1-bad round. Let the two agents corresponding to this round G_i be (g_i, y). So, $\Big(\texttt{FAIRLOC}(\mathcal{T}_i)\Big)^n \leq 3 \cdot \Big(\mathcal{OPT}_{assign}(\mathcal{T}_i)\Big)^{2n} \cdot \vartheta$. Here $\vartheta = 2 \sup_{x,y \in \mathcal{X}} d(x, y)$ is the diameter of the feature space. Thus, this leads to $\texttt{FAIRLOC}(\mathcal{T}_i) \leq 3^{1/n} \Big(\mathcal{OPT}_{assign}(\mathcal{T}_i)\Big)^2 \vartheta^{1/n}$.

Bounding 2-bad rounds: First, assume that there are even number of 2-bad rounds. In this case consider the pairs of consecutive 2-bad rounds as $G_i : (g_i, h_i)$ and $G_i^{'} = (g_i', h_i')$ with $G_i^{'}$ bad round followed by G_i (Figure (4.1b)). Note that $g_i, g_i' \in \mathcal{C}_2$ and $h_i, h_i' \in \mathcal{C}_1$. Now consider instance $\mathcal{T}_i = \{C, \{g_i, g_i', h_i, h_i'\}\}$, then , $\mathcal{OPT}_{assign}(\mathcal{T}_i) = d(h_i, f_1) \cdot d(h_i', f_1) \cdot d(g_i, f_2) \cdot d(g_i', f_2)$. As a consequence of the allocation rule used by FAIRLOC we have

$$\begin{aligned}
&d(g_i, f_1) \leq d(h_i, f_1),\ d(g_i', f_1) \leq d(h_i', f_1), d(h_i, f_2) \leq d(g_i', f_2)\\
&\text{and } d(h_i, f_2) \leq d(h_i', f_2).
\end{aligned} \tag{4.21}$$

Furthermore,

$$\begin{aligned}
\texttt{FAIRLOC}(\mathcal{T}_i) &= d(g_i, f_1) \cdot d(g_i', f_1) \cdot d(h_i, f_2) \cdot d(h_i', f_2) && (4.22)\\
&\leq d(h_i, f_1) \cdot d(h_i', f_1) \cdot d(g_i', f_2) \cdot d(h_i', f_2) && \text{(using Equation 4.21)}\\
&\leq d(h_i, f_1) \cdot d(h_i', f_1) \cdot d(g_i', f_2) \cdot \Big(d(h_i', f_1) + d(f_1, f_2)\Big) && (4.23)\\
& && (\because \text{triangle inequality})
\end{aligned}$$

$$\leq d(h_i, f_1) \cdot d(h_i', f_1) \cdot d(g_i', f_2) \cdot \Big(d(h_i', f_1) + d(g_i, f_1) \tag{4.24}$$

$$+ d(g_i, f_2) \Big) \qquad (\because \text{triangle inequality})$$

$$\leq d(h_i, f_1) \cdot d(h_i', f_1) \cdot d(g_i', f_2) \cdot \Big(d(h_i', f_1) + d(h_i, f_1) \tag{4.25}$$

$$+ d(g_i, f_2) \Big) \qquad (\text{using Equation 4.21})$$

$$\leq 2 \cdot \Big(\mathcal{OPT}_{assign}(\mathcal{T}_i) \Big)^{2n} + \Big(\mathcal{OPT}_{assign}(\mathcal{T}_i) \Big)^{n} \leq 3 \cdot \Big(\mathcal{OPT}_{assign}(\mathcal{T}_i) \Big)^{2n} \tag{4.26}$$

If there are odd number of 2-bad rounds then, let $G = (g_i, h_i)$ be the last 2-bad round. It is easy to see that $(\texttt{FAIRLOC}(\mathcal{T}_i))^n \leq 3 \cdot (\mathcal{OPT}_{assign}(\mathcal{T}_i))^{2n} \cdot d(g_i, f_2) \cdot d(h_i, f_1) \leq 3 \cdot (\mathcal{OPT}_{assign}(\mathcal{T}_i))^{2n} \cdot \vartheta^2$. So, $\texttt{FAIRLOC}(\mathcal{T}_i) \leq 3^{1/n} \cdot \Big(\mathcal{OPT}_{assign}(\mathcal{T}_i) \Big)^{2} \cdot \vartheta^{2/n}$. Here $\vartheta = 2 \sup_{x,y \in \mathcal{X}} d(x, y)$ is the diameter of the feature space.
Thus, since each $\mathcal{T}_i$ instance has distinct agent locations (or data points), we can get the overall bound as follows where r_1, r_2 is the total number of 1-bad and 2-bad rounds respectively :

$$\texttt{FAIRLOC}(\mathcal{T}) = \begin{cases} \prod_{i=1}^{r} \texttt{FAIRLOC}(\mathcal{T}_i) & \text{if both } r_1, r_2 \text{ is even} \\ \prod_{i=1}^{r_1} \texttt{FAIRLOC}(\mathcal{T}_i) \prod_{j=1}^{r_2} \texttt{FAIRLOC}(\mathcal{T}_j) \cdot \vartheta^{2/n} & \text{if } r_1 \text{ is even and } r_2 \text{ is odd} \\ \prod_{i=1}^{r_1} \texttt{FAIRLOC}(\mathcal{T}_i) \cdot \vartheta^{1/n} \prod_{j=1}^{r_2} \texttt{FAIRLOC}(\mathcal{T}_j) & \text{if } r_1 \text{ is odd and } r_2 \text{ is even} \\ \prod_{i=1}^{r_1} \texttt{FAIRLOC}(\mathcal{T}_i) \cdot \vartheta^{1/n} \prod_{j=1}^{r_2} \texttt{FAIRLOC}(\mathcal{T}_j) \cdot \vartheta^{2/n} & \text{if } r_1 \text{ is odd and } r_2 \text{ is odd} \end{cases} \tag{4.27}$$

$$\leq \prod_{i=1}^{\lfloor r = r_1 + r_2 \rfloor} 3^{1/n} \cdot (\mathcal{OPT}_{assign}(\mathcal{T}_i))^2 \cdot \vartheta^{3/n} \leq 3^{1/4} \cdot \Big(\mathcal{OPT}_{assign}(\mathcal{T}) \Big)^{2} \cdot \vartheta^{3/4}$$

$(\because r \leq n/4$ as each instance consumes four agents$)$

□

It is important to note that since we are providing cost bounds on the product of distances, achieving a quadratic bound in terms of optimal is not bad. The bounds intuitively convey the idea that the results are two approximations for the logarithmic sum objective formulation of Nash social welfare. Similar bounds in terms of optimal are also evident in Nash social welfare works for resource allocation [219, 220].

Corollary 4.6. *For $k = 2$ and $\tau_\ell = \frac{1}{k}$ for all $\ell \in [m]$, we have $\texttt{FAIRLOC}(\mathcal{I}) \leq 3^{1/4} \cdot ((2^n + 1) \cdot \beta) \cdot \vartheta^{3/4} \cdot (\mathcal{OPT}_{FLP}(\mathcal{I}))^4$-approximate where β is approximation factor for vanilla facility location problem for any given instance $\mathcal{I}$.*

The above corollary is a direct consequence of Lemma 4.1 and the fact that $\texttt{FAIRLOC}(F, X) \leq \texttt{FAIRLOC}(\hat{F}, X)$. Here, $\hat{F}, F$ are centers of vanilla allocation and fair allocation obtained by FAIRLOC respectively. The bounds can easily be extended for k facilities along similar lines by looking at cycles of length at most $(k-1)$ to directly obtain 2^{k-1}-approximate solution with respect to $\boldsymbol{\tau}$-ratio fair assignment problem. The final result obtained is as follows:

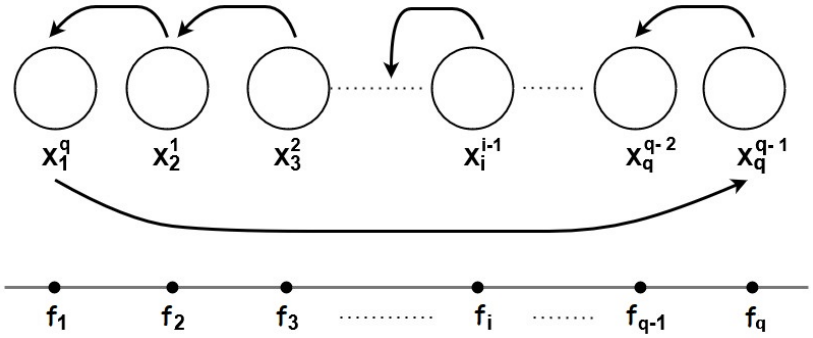


Figure 4.2: Visual representation of set X_i^j and cycle of length q for Theorem 4.7. The arrow represents the direction from the assigned facility to the facility in optimal allocation. Thus, for each set X_i^j we have f_i as the currently assigned facility and f_j as the facility in optimal assignment.

Theorem 4.7. *When $\tau_\ell = \frac{1}{k}$ for all $\ell \in [m]$, an allocation returned by FAIRLOC for given facility and agent locations is $\boldsymbol{\tau}$-ratio fair and satisfies $(3^{1/4}\vartheta^{3/4})^{2^{k-1}}(\mathcal{OPT}_{assign}(\mathcal{T}))^{2^{k-1}}$-approximation to the product of distances objective, i.e., 2^{k-1}-approximation guarantee to logarithmic sum of distances with respect to an optimal $\boldsymbol{\tau}$-ratio fair assignment up to an instance-dependent multiplicative constant.*

Proof. In the previous proof, we basically considered two length cycles. Two 1-bad allocations resulted in one type of cycle, and one 2-bad allocations resulted in another type of cycle. When the number of facilities is greater than two, then any $2 \leq q \leq k$ length cycles can be formed. Without loss of generality, let us denote $\{f_1, f_2, \ldots, f_q\}$ as the centers that are involved in forming such cycles. Further denote by set X_i^j to be the set of agent locations that are allotted to facility i by FAIRLOC but should have been allotted to facility j in an optimal fair allocation. The q length cycle can then be visualized in Figure 4.2 with the arrow pointing towards the optimal facility. As the cycle is formed with respect to these agents, we have $|X_1^q| = |X_2^1| = \ldots = |X_q^{q-1}|$ The cost by FAIRLOC algorithm is then given as:

$$(\texttt{FAIRLOC}(\mathcal{T}))^n = \prod_{i=2}^{q} \prod_{x \in X_i^{i-1}} d(x, f_i) \cdot \prod_{x \in X_1^q} d(x, f_1) \tag{4.28}$$

$$\leq 3^{n/4} \left(\prod_{x \in X_2^1} (d(x, f_1))^2 \cdot \prod_{x \in X_1^q} (d(x, f_2))^2 \cdot \vartheta^{3n/4} \right) \cdot \prod_{i=3}^{q} \prod_{x \in X_i^{i-1}} d(x, f_i)$$

(using Theorem 4.4)

$$\leq 3^{n/4}\left(\prod_{x\in X_2^1}(d(x,f_1))^2\cdot\vartheta^{3n/4}\right)\cdot\left(\prod_{x\in X_1^q}(d(x,f_2))\right)^2\cdot\prod_{i=4}^{q}\prod_{x\in X_i^{i-1}}d(x,f_i) \tag{4.29}$$

$$\leq 3^{n/4}\cdot\vartheta^{3n/4}\cdot\left(\prod_{x\in X_2^1}(d(x,f_1))\right)^2\cdot\left(\prod_{x\in X_1^q}(d(x,f_2))\right)^2\cdot\left(\prod_{x\in X_2^3}(d(x,f_3))\right)$$
$$\cdot\left(\prod_{i=4}^{q}\prod_{x\in X_i^{i-1}}d(x,f_i)\right) \tag{4.30}$$

$$\leq 3^{n/4}\cdot\vartheta^{3n/4}\cdot\left(\prod_{x\in X_2^1}(d(x,f_1))\right)^2\cdot\left(\prod_{x\in X_1^q}(d(x,f_2))\prod_{x\in X_2^3}(d(x,f_3))\right)^2\cdot\prod_{i=4}^{q}\prod_{x\in X_i^{i-1}}d(x,f_i) \tag{4.31}$$

$$\leq 3^{n/4}\cdot\vartheta^{3n/4}\cdot\left(\prod_{x\in X_2^1}(d(x,f_1))\right)^2\cdot\left(3^{n/4}\prod_{x\in X_3^2}(d(x,f_2))^2\cdot\prod_{x\in X_1^q}(d(x,f_3))^2\cdot\vartheta^{3n/4}\right)^2$$
$$\cdot\left(\prod_{i=4}^{q}\prod_{x\in X_i^{i-1}}d(x,f_i)\right) \qquad \text{(using Theorem 4.4)}$$

$$\leq 3^{\frac{n}{4}+\frac{2n}{4}}\cdot\vartheta^{\frac{3n}{4}+\frac{6n}{4}}\cdot\left(\prod_{x\in X_2^1}(d(x,f_1))\right)^2\cdot\left(\prod_{x\in X_3^2}(d(x,f_2))^2\cdot\prod_{x\in X_1^q}(d(x,f_3))^2\right)^2\cdot$$
$$\cdot\prod_{i=4}^{q}\prod_{x\in X_i^{i-1}}d(x,f_i) \tag{4.32}$$

$$\leq 3^{\frac{n}{4}+\frac{2n}{4}}\cdot\vartheta^{\frac{3n}{4}+\frac{6n}{4}}\cdot\left(\prod_{x\in X_2^1}(d(x,f_1))\right)^2\cdot\left(\prod_{x\in X_3^2}(d(x,f_2))\right)^4\cdot\left(\prod_{x\in X_1^q}(d(x,f_3))\right)^4$$
$$\cdot\left(\prod_{x\in X_4^3}(d(x,f_4))\right)\cdot\left(\prod_{i=5}^{q}\left(\prod_{x\in X_i^{i-1}}d(x,f_i)\right)\right) \tag{4.33}$$

$$\leq 3^{\frac{n}{4}+\frac{2n}{4}}\cdot\vartheta^{\frac{3n}{4}+\frac{6n}{4}}\cdot\left(\prod_{x\in X_2^1}(d(x,f_1))\right)^2\cdot\left(\prod_{x\in X_3^2}(d(x,f_2))\right)^4\cdot\left(\prod_{x\in X_1^q}(d(x,f_3))\right.$$
$$\left.\cdot\prod_{x\in X_4^3}(d(x,f_4))\right)^4\cdot\left(\prod_{i=5}^{q}\left(\prod_{x\in X_i^{i-1}}d(x,f_i)\right)\right) \tag{4.34}$$

$$\leq 3^{\frac{n}{4}+\frac{2n}{4}}\cdot\vartheta^{\frac{3n}{4}+\frac{6n}{4}}\cdot\left(\prod_{x\in X_2^1}(d(x,f_1))\right)^2\cdot\left(\prod_{x\in X_3^2}(d(x,f_2))\right)^4\cdot\left(3^{n/4}\cdot\prod_{x\in X_4^3}(d(x,f_3))^2\right.$$
$$\left.\cdot\prod_{x\in X_1^q}(d(x,f_4))^2\cdot\vartheta^{3n/4}\right)^4\cdot\left(\prod_{i=5}^{q}\prod_{x\in X_i^{i-1}}d(x,f_i)\right) \qquad \text{(using Theorem 4.4)}$$

$$\leq 3^{\frac{n}{4}+\frac{2n}{4}+\frac{4n}{4}}\cdot\vartheta^{\frac{3n}{4}+\frac{6n}{4}+\frac{12n}{4}}\cdot\left(\prod_{x\in X_2^1}(d(x,f_1))\right)^2\cdot\left(\prod_{x\in X_3^2}(d(x,f_2))\right)^4\cdot\left(\prod_{x\in X_4^3}(d(x,f_3))\right. \tag{4.35}$$

$$\cdot \prod_{x \in X_1^q} (d(x, f_4))\Bigg)^8 \cdot \left(\prod_{i=5}^{q} \prod_{x \in X_i^{i-1}} d(x, f_i)\right)$$

$$\leq \left(3^{n/4} \cdot \vartheta^{3n/4}\right)^{2^{q-1}} \left(\mathcal{OPT}_{assign}(\mathcal{T})\right)^{n \cdot 2^{q-1}} \tag{4.36}$$

$$\implies (\texttt{FAIRLOC}(\mathcal{T}))^n \leq \left(3^{n/4} \cdot \vartheta^{3n/4}\right)^{2^{q-1}} \left(\mathcal{OPT}_{assign}(\mathcal{T})\right)^{n \cdot 2^{q-1}} \tag{4.37}$$

$$\implies \texttt{FAIRLOC}(\mathcal{T}) \leq \left(3^{1/4} \cdot \vartheta^{3/4}\right)^{2^{q-1}} \left(\mathcal{OPT}_{assign}(\mathcal{T})\right)^{2^{q-1}} \tag{4.38}$$

$$\implies \texttt{FAIRLOC}(\mathcal{T}) \leq \left(3^{1/4} \cdot \vartheta^{3/4}\right)^{2^{k-1}} \left(\mathcal{OPT}_{assign}(\mathcal{T})\right)^{2^{k-1}} \quad (\because q \leq k) \tag{4.39}$$

Here, the first inequality follows by exchanging the agents in X_2^1 and X_1^q using Theorem 4.4. As the maximum length cycle possible is k, we straight away get the proof of 2^{k-1}-approximation to logarithmic sum of product of distances objective. □

The above theorem shows that FAIRLOC achieves an exponential approximation in terms of k, but our experimental observations on Nash objective do not degrade too much with increasing k and follows along the lines as Theorem 4.4. We leave this as a future study to prove tight approximation guarantees and now provide cost guarantees with respect to the general $\boldsymbol{\tau}$ vector for k=2.

4.5.1 Guarantees for FAIRLOC for general τ

Given an instance $\mathcal{T}$, facility opening locations F, and set of agents X, we start with a simple observation that problem of solving $\boldsymbol{\tau}$-ratio fair assignment can be divided into two subproblems:

1. Solving optimal $1/k$-ratio fair assignment problem on subset of agents $X_1 \in X$ such that $|X_1| = \sum_{\ell \in [m]} k\tau_\ell n_\ell$.
2. Solving optimal fair assignment problem on $X_2 \in X \setminus X_1$ without any fairness constraint.

Let us denote the first instance by $\mathcal{T}^{1/k}$ and second instance with $\mathcal{T}^0$, i.e. $\mathcal{T}^{1/k} = \{X_1, F\}$ and $\mathcal{T}^0 = \{X_2, F\}$.

Lemma 4.8. *There exists two separate instances $\mathcal{T}^{1/k}$ with τ=$\{1/k\}_{\ell=1}^m$ and $\mathcal{T}^0$ with τ=$\{0\}_{\ell=1}^m$ such that solving the fair assignment problem on instance $\mathcal{T}$ can be divided into solving fair assignment on these two instances, i.e., $\mathcal{OPT}_{assign}(\mathcal{T}) = \mathcal{OPT}_{assign}(\mathcal{T}^{1/k}) \cdot \mathcal{OPT}_{assign}(\mathcal{T}^0)$.*

Proof. The $\mathcal{T}$ instance requires that each facility should have at least τ_ℓ n_ℓ number of agents for each protected group value. The remaining agents can be allocated in an

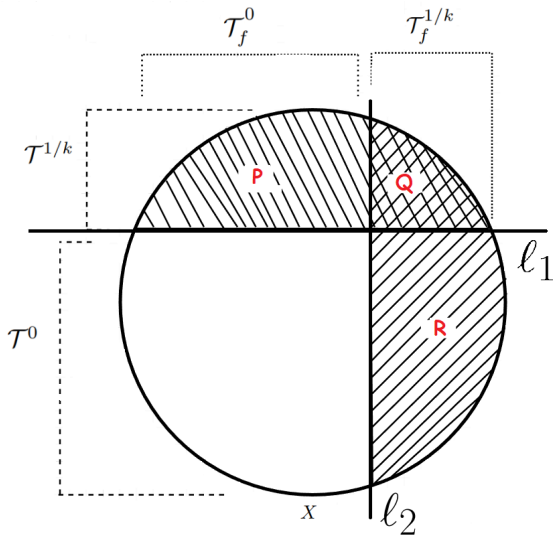


Figure 4.3: Set of agents X divided into instance $\mathcal{T}^{1/k}$ and $\mathcal{T}^0$. Further the instances $\mathcal{T}_f^{1/k}$ and $\mathcal{T}_f^0$ are depicted in the same set of agents X leading to formation of regions P, Q, R.

optimal manner without any fairness constraint. Therefore in an optimal assignment, there exists a set X_1^{OPT} such that $|X_1^{OPT}| = \sum_{\ell=1}^{m} \tau_\ell n_\ell k$ that satisfies the $\tau-$ratio fairness with $\tau_\ell = 1/k \ \forall \ell \in [m]$. □

Let X_1^f be the set of agents that are allocated in lines 4-13 by Algorithm 4 (fair procedure). Further, let $\mathcal{T}_f^{1/k}$ be an instance to τ-ratio fair assignment problem with $\tau = \{1/k\}_{\ell=1}^m$ and consisting of agents X_1^f and $\mathcal{T}_f^0$ be instance when $\tau=\{0\}_{\ell=1}^m$ by `FAIRLOC` (depicted in Figure 4.3). Then, our next lemma shows that the partition returned by `FAIRLOC` is the optimal one.

Lemma 4.9. *$\mathcal{OPT}_{assign}(\mathcal{T}_f^{1/k}) \cdot \mathcal{OPT}_{assign}(\mathcal{T}_f^0) \leq \mathcal{OPT}_{assign}(\mathcal{T}^{1/k}) \cdot \mathcal{OPT}_{assign}(\mathcal{T}^0)$ for any partition $\mathcal{T}^{1/k}$ and $\mathcal{T}^0$. Thus, $\mathcal{OPT}_{assign}(\mathcal{T}) = \mathcal{OPT}_{assign}(\mathcal{T}_f^{1/k}) \cdot \mathcal{OPT}_{assign}(\mathcal{T}_f^0)$.*

Proof. The proof of this lemma follows same lines as proof of Lemma 3.20. □

Theorem 4.10. *For k=2 and any general τ vector, an allocation returned by `FAIRLOC` guarantees $\boldsymbol{\tau}$-ratio fairness and satisfies $3^{1/4} \cdot \vartheta^{3/4} \cdot (\mathcal{OPT}_{assign}(\mathcal{T}))^2$-approximate guarantee with respect to an fair assignment problem.*

Proof. With the help of Lemma 4.8 the cost of `FAIRLOC` on instance $\mathcal{T}_f$ can be computed as,

$$\texttt{FAIRLOC}(\mathcal{T}) = \texttt{FAIRLOC}(\mathcal{T}_f^{1/k}) + \texttt{FAIRLOC}(\mathcal{T}_f^0) \tag{4.40}$$

Now, from Equation 4.27, $\texttt{FAIRLOC}(\mathcal{T}_f^{1/k}) \leq 3^{1/4} \left(\mathcal{OPT}_{assign}(\mathcal{T}_f^{1/k}) \right)^2 \vartheta^{3/4}$.

Also, as $\mathcal{T}_f^0$ is solved for $\tau=\{0\}_{\ell=1}^m$ i.e. assignment is carried solely on the basis of k-means allocation, we have $\texttt{FAIRLOC}(\mathcal{T}_f^0) = \mathcal{OPT}_{assign}(\mathcal{T}_f^0) \leq 3^{1/4} \cdot (\mathcal{OPT}_{assign}(\mathcal{T}_f^0))^2$.

Equation 4.40 becomes,

$$\texttt{FAIRLOC}(\mathcal{T}) \leq 3^{1/4} \cdot (\mathcal{OPT}_{assign}(\mathcal{T}_f^{1/k}))^2 \cdot \vartheta^{3/4} + (\mathcal{OPT}_{assign}(\mathcal{T}_f^0))^2 \tag{4.41}$$

$$\leq 3^{1/4} \cdot \vartheta^{3/4} \cdot \left((\mathcal{OPT}_{assign}(\mathcal{T}))^2 \cdot (\mathcal{OPT}_{assign}(\mathcal{T}_f^0))^2 \right) \quad \text{(using Lemma 4.8)}$$

$$\leq 3^{1/4} \cdot \vartheta^{3/4} \cdot (\mathcal{OPT}_{assign}(\mathcal{T}))^2 \tag{4.42}$$

□

The proof for the case of general $\boldsymbol{\tau}$ vector for any number of facilities follows the same lines and results in same approximation as Theorem 4.7.

4.6 Experimental Results and Analysis

We now validate the efficacy of proposed FAIRLOC on real-world United States (US) census dataset for agent population with various observable facilities captured in Homeland Infrastructure Foundation-Level (HIF) Data by the US Dept of Homeland Security. The population data released by US Census Planning captures information about the demographic composition of agents residing in states of the US during 2022. The complete dataset is publicly accessible using census-downloader API[1]. The data records the census tract level distribution of different protected group values, and we mainly divide the complete United States into four popular regions as follows and illustrated[2] in Figure 4.4:

(1) **US North**: It primarily includes the states of New York, Massachusetts, Pennsylvania, and nearby states, making a total of nine states. The complete preprocessing script is provided as a GitHub repository[3]. The region encompasses a total of 42.6 million agents across 18,000 tracts, and we randomly subsample 100 census tracts to avoid high computation and memory.

(2) **US South**: includes the states of Texas, Florida, Georgia, and other 13 states with an agent population of 74 million, and we sample 100 tracts randomly spread across the region.

(3) **US West**: It is made up of popular states of Washington D.C., California, Arizona and others, making a total of 13 states with 53 million agents. For the present study, we consider the Pacific US to be a part of the West and a total of 100 tracts.

(4) **US MidWest**: comprises the regions around Wisconsin, Indiana, and Illinois. It contains a spread of 55.2 million agents across 15000 tracts, and we use 100 tracts across a total of 12 states. The total number of agents in each tract depends on the protected group chosen and is summarized in Table 1.

Next, the **facility dataset records** information about the facility's existing real-world

[1]https://github.com/datadesk/census-data-downloader

[2]The figure is royalty-free download preview provided by dreamstime.com

[3]https://github.com/fairloc/code

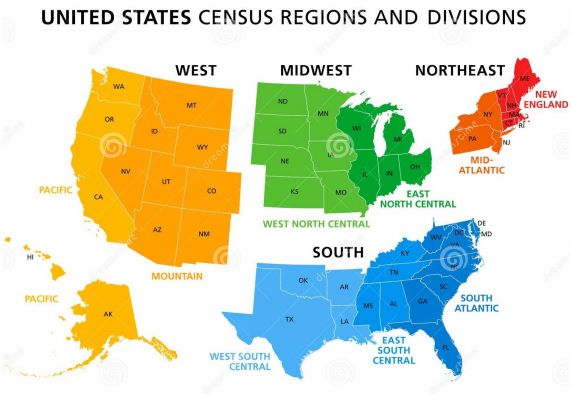


Figure 4.4: The figure shows census regions by United States Census Planning. We consider the US-Pacific as part of the US-West.

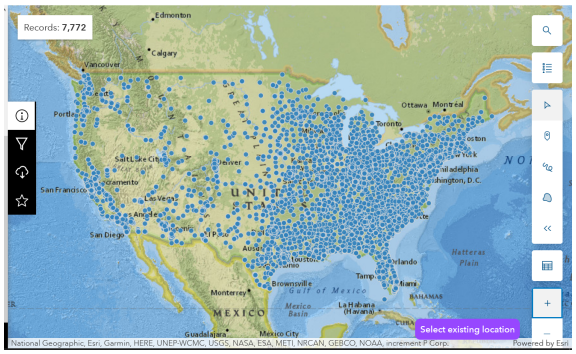


Figure 4.5: United States map view of dialysis centers in Homeland Infrastructure Foundation dataset.

locations in different United States regions. The facility under consideration in the present study is as follows:

1. **Dialysis Centers**: The dataset was primarily curated with the assistance of Rx, an organization that helps patients locate nearby kidney care centers. The publicly available records at HIF include data from Rx and other pharmaceutical companies. The dataset contains information on approximately $7,772$ dialysis centers across the United States. A map view of these facilities from the GeoPlatform ArcGIS Online portal is shown in Figure 4.5.

2. **Public Schools**: The data contains information (latitude and longitude) about all public elementary and secondary education schools as facilities in the US. The data is updated up to the 2022 session and contains about $102,268$ records.

3. **Pharmacy**: Consists of a total of $63,018$ pharmacy locations spread across the US.

4. **National Shelter System (NSS)** : includes information on the latitude and longitude of facilities that can accommodate agents during disaster emergencies and evacuations. It consists of $68,934$ locations approved by the Federal Emergency Management Agency (FEMA).

Data Preprocessing: The datasets need a certain level of pre-processing before making them ready for use to validate the efficacy of the algorithm. Firstly, although the United States census population dataset contains information about different protected group values at each census tract, the latitude and longitude coordinates of tracts are not directly available in the public dataset. One needs to map the census tract ID (called Geoid) with the corresponding latitude-longitude information available in TIGER Shapefiles provided by the Census Bureau. Secondly, we divide the complete datasets for both agent population and facilities into four parts based on regions- North, South, West and Midwest. The code for the preprocessing, `FAIRLOC` and datasets are available as GitHub repository[4]. We next look into the protected groups that are under scrutiny in the present paper:

- **Poverty level** (P): Binary group indicating agent's above poverty line status. Used for analyzing pharmacy facilities. The protected group takes two values: whether the income level has been below poverty in the past 12 months or above.

- **Income level** (I): A quartet (four) valued protected group used in dialysis, schools and NSS records. It can take values from buckets: $0-24999$, $25000-74999$, $75000-199999$ and $200000+$.

- **Age-Gender** (AG): Categorizes age (0 to 65+) and gender pairs into ten buckets. It is used in NSS facility datasets. The protected group buckets are Male (0-17), Female (0-17), Male (18-34), Female (18-34), Male (35-49), Female (35-49), Male (50-64), Female (50-64), Male (65+), Female (65 and plus).

- **Race** (R): A quintet (five) racial regions used in the dialysis dataset. The five buckets are White, Black, Latino, Asian and others.

- **Language** (L): A quintet (five) group used in pharmacy and school facility analysis. The group takes value - only English, Only Spanish, other Indo-European, Asian and pacific islander languages and finally, others.

Having looked into all possible regions, facilities and protected groups we summarize in Table 4.1 the total number of agents subsampled in 100 census tracts in all the settings.

4.6.1 Comparison against Different τ Vectors

We compare the performance of `FAIRLOC` on a fixed number of facilities, i.e., k=10 for all regions and facilities from Figure 4.9 to 4.28. In this study, we evaluate different metrics provided below for different user-desired levels of group fairness ($\boldsymbol{\tau}$ vectors). We consider $\tau_\ell = 0\ \forall \ell \in [m]$, represented by $\boldsymbol{\tau}_0$, which does not consider any group fairness constraints.

[4]https://github.com/fairloc/code

Region	Facility	Protected Group	No of Agents
West	Dialysis	Income	154021
	Dialysis	Race	355527
	NSS	Age-Gender	409481
	Pharmacy	Poverty level	406700
	Schools	Language	144312
Midwest	Dialysis	Income	146174
	Dialysis	Race	358445
	NSS	Age-Gender	339502
	Pharmacy	Poverty level	326205
	Schools	Language	146154
South	Dialysis	Income	138076
	Dialysis	Race	380974
	NSS	Age-Gender	371266
	Pharmacy	Poverty level	359542
	Schools	Language	132340
North	Dialysis	Income	149643
	Dialysis	Race	405159
	NSS	Age-Gender	372233
	Pharmacy	Poverty level	361391
	Schools	Language	152002

Table 4.1: The table describes the total number of agents in each region for given facility and protected group pairs.

4.6.2 Performance Metrics

1. **Utilitarian Cost** (↓): is the sum of the distances of agents to their assigned facility (Definition 4.2).

2. **Nash Value** (↑) [221]: The goal of Nash social welfare is to maximize the product of utilities (opposite to cost). To this, we measure Nash value as $\prod_{x_i \in X} \log(\Delta - d(x_i, f))$ where $\Delta = 1 + \max_{(x_i,f)\in(X,F)} d(x_i, f)$.

We evaluate efficacy on following **group fairness metrics**:

1. **Balance** (↑) [42]: is defined as the minimum ratio of dominant to minority protected group agents across all facilities.

2. **Fairness Error** (↓) [12]: measures the KL-divergence between the achieved proportion of different group type of agent (denoted as P_f^ℓ for type ℓ) at each facility ($f \in F$) to desired user input vector $\boldsymbol{\tau}$ i.e., $\sum_{f\in F} \sum_{\ell\in[m]} -\tau_\ell \log(P_f^\ell)$.

Since optimizing the Nash social welfare prefers assignments that are more equitable, we validate these findings with metrics motivated by individual fairness in facility location problem [88, 222]. The metric captures each agent's expectations of distance to facilities by constructing regional density balls around them with fair radius $\alpha \cdot r(x)$ (Definition 4.1).

Here, α is the approximation factor to Individual Fairness (IF). The lower value of α indicates that the agent does not need to travel too far and finds a facility within its local density. The prior works [88, 222] in individual fair facility location problem theoretically

provide 2 and 6-approximation guarantees, respectively, when k or fewer facilities need to be opened. However, their algorithmic design needs hyper-parameter tuning in case exact k facilities need to be opened and thus practically results in higher α values. We carefully analyze the distribution of α values using the following **individual fairness metrics**- **(1) Mean value** ($\downarrow$) **(2) Median value** ($\downarrow$) and **(3) Maximum value** ($\downarrow$) of α across agents. Since Jung et al. [88] provides 2-approximation results to fair radius, we record **(4) Fraction of agents having** $\alpha \leq 2$ ($\uparrow$) for comparison.

4.6.3 Baselines

1. **Jung** Jung et al. [88]: focuses on optimizing facility locations while ensuring individual fairness. Authors further employ binary search for parameter tuning to open exactly (or close to) k centers. Note that the method assumes that facilities can only be opened at the agent's location, so we execute the method by setting $F = X$ and then map each suggested location $f^{'} \in X$ to the closest $f \in F$.

2. **LSPP** Bateni et al. [222]: improvises the algorithm proposed by [88] by applying a swapping-based local-search method and requires tuning a few parameters to open exactly k centers. LSPP also assumes $F = X$, so we modify it as discussed earlier.

3. **FRAC**$_{OE}$ Shivam Gupta et al. [44]: is a post-processing method that modifies the unfair clustering assignments to strictly satisfy $\boldsymbol{\tau}$-ratio fairness. We overcome the assumption of $F = X$ as we did in Jung and LSPP.

4. **Jung+RR**: This method considers the baseline algorithm as Jung to satisfy individual fairness and then applies FRAC$_{OE}$ to obey group fairness.

4.6.4 Analysis on Varying τ Vectors

We first compare the performance of `FAIRLOC` on a fixed number of facilities, i.e., k=10 for all regions and facilities. In this study, we evaluate different metrics for different user-desired levels of group fairness ($\boldsymbol{\tau}$ vectors). We consider $\tau_\ell = 0\ \forall \ell \in [m]$, represented by $\boldsymbol{\tau}_0$, which does not consider any group fairness constraints. It helps us evaluate the results for vanilla k-means variation of FRAC$_{OE}$, which acts as a lower bound on the utilitarian cost. Other four $\boldsymbol{\tau}$ vectors were randomly generated to cover various scenarios and are listed as follows. The value 0.1/4 means at least 0.025% fraction of total agents from that group value assigned to each facility.

Income Levels

$$\boldsymbol{\tau}_1 = [0.1/4,\ 0.1/4,\ 0.1/4,\ 0.1/4];; \quad \boldsymbol{\tau}_2 = [0.1/4,\ 0.1/8,\ 0.1/8,\ 0.005];$$
$$\boldsymbol{\tau}_3 = [0.005,\ 0.005,\ 0.031,\ 0.1/8];; \quad \boldsymbol{\tau}_4 = [0.1/4,\ 0.1/8,\ 0.04,\ 0.005];$$

Race and Language

$$\boldsymbol{\tau}_1 = [0.1/4,\ 0.1/4,\ 0.1/4,\ 0.1/4,\ 0.1/4]; \quad \boldsymbol{\tau}_2 = [0.1/4,\ 0.1/4,\ 0.1/8,\ 0.1/8,\ 0.005];;$$
$$\boldsymbol{\tau}_3 = [0.005,\ 0.005,\ 0.031,\ 0.04,\ 0.1/8]; \quad \boldsymbol{\tau}_4 = [0.1/4,\ 0.1/8,\ 0.04,\ 0.005,\ 0.031];$$

Age-Gender

$$\boldsymbol{\tau}_1 = [0.1/4,\ 0.1/4,\ 0.1/4,\ 0.1/4,\ 0.1/4,\ 0.1/4,\ 0.1/4,\ 0.1/4,\ 0.1/4,\ 0.1/4];$$
$$\boldsymbol{\tau}_2 = [0.1/4,\ 0.1/4,\ 0.1/4,\ 0.1/8,\ 0.1/8,\ 0.1/8,\ 0.005,\ 0.005,\ 0.031,\ 0.031];$$
$$\boldsymbol{\tau}_3 = [0.005,\ 0.005,\ 0.005,\ 0.005,\ 0.005,\ 0.1/8,\ 0.1/8,\ 0.1/8,\ 0.031,\ 0.031];$$
$$\boldsymbol{\tau}_4 = [0.1/4,\ 0.1/4,\ 0.1/4,\ 0.1/8,\ 0.1/8,\ 0.1/8,\ 0.04,\ 0.04,\ 0.031,\ 0.031];$$

Poverty Levels

$$\boldsymbol{\tau}_1 = [0.1/4,\ 0.1/4]; \quad \boldsymbol{\tau}_2 = [0.1/4,\ 0.1/8]; \quad \boldsymbol{\tau}_3 = [0.005,\ 0.031]; \quad \boldsymbol{\tau}_4 = [0.04,\ 0.031];$$

The results are provided from Figure 4.9 to 4.28 and summarized below:
Low Utilitarian Cost of `FAIRLOC`: `FAIRLOC` gives low utilitarian cost across all $\boldsymbol{\tau}$ vectors. We see a high utilitarian cost for algorithms like FRAC_{OE}, which specifically focuses on optimizing utilitarian cost because `FAIRLOC` optimizes over facility set F. On the other hand, output facilities of FRAC_{OE} are modified by mapping each output facility to the closest facility in F. The amplification in cost is also due to considering the actual road map distance to locate the closest facility, which does not satisfy triangular inequality, and the base algorithm k-means in FRAC_{OE} heavily relies on the properties of p-norm distance metrics.
Observations on Nash value: Nash values for `FAIRLOC` are slightly higher for initial $\boldsymbol{\tau}$ vectors representing relaxed group fairness constraints. It is important to note that since the Nash value metric involves logarithmic terms, it compresses large values. The gap becomes more clear from the results in the next varying k experiments. For stricter $\boldsymbol{\tau}$ vectors, we observe that Nash values are close enough across all benchmarks due to more reassignments to satisfy group fairness.
Observations on Fairness: (1) `FAIRLOC` does significantly better on the balance and fairness error, showing its efficacy across settings. The results degrade in Jung, LSPP for balance, thus leading to high fairness errors. Though post-processing helps improve balance in Jung+RR, the performance is still less effective than `FAIRLOC`. (2) In individual fairness metrics, FRAC_{OE} has a considerably low fraction of individual fairness agents and higher α statistics. Note that since Jung and LSPP need tuning for opening exactly k facilities, the theoretical results no longer hold, resulting in higher α values and deviation. On the contrary, `FAIRLOC` stands out and helps achieve individual fairness inherently by its equitable design (Nash) while obeying group fairness.

4.6.5 Analysis across Varying k

In this experiment, we now measure the performance of different methods on increasing k. The results for k as $2, 5, 10, 25$ and 30 are provided in Figure 4.29 to 4.48 at fixed $\boldsymbol{\tau}_2$ and summarized below.

Observations: (1) `FAIRLOC` maintains its efficacy on cost, Nash value across k. Jung's performance improves after a certain threshold k as opening higher facilities assists in better tuning. However, both Jung and LSPP suffer in group fairness metrics as they are not designed to handle it.

(2) Note that the mean α is slightly close to Jung+RR and higher than Jung at increasing k. But the performance on the median is good. The main reason is that mean values are subject to outliers and can verify that the maximum α value increases at higher k owing to dataset characteristics. But still, the efficacy of `FAIRLOC` is validated by the fact that % of individually fair (IF) agents are among the highest in our algorithm and at a larger gap than Jung and Jung+RR. Thus, `FAIRLOC` stands out as one achieving lower costs while simultaneously obeying group and individual fairness.

4.6.6 Ablation Study on `FAIRLOC`

Analysis on facility ordering in `FAIRLOC`'s round-robin

We first show that the `FAIRLOC` performance is invariant to the random ordering or round robin. To this, we compute the variance on Nash value across 100 random permutations for the NSS dataset in all regions at fixed $\boldsymbol{\tau}_2$, $k = 10$. We observe that the variance is of orders 10^4 compared to Nash value having orders 10^6. Considering the size of the dataset ($\sim 10^5$ order), we can say `FAIRLOC` is invariant to ordering (as $10^4 << 10^6$).

Distribution of α values across agents

We plot the histogram bins of different α values and observe that the plots are right skewed for `FAIRLOC` showcasing its efficacy in satisfying equitable allocation to all agents. Thus, we can say that minimizing the product of distances helps achieve a facility center within a good approximation to the fair radius for most of the agents. Notably, on comparing the distribution of fair setting with $\tau_\ell = 0\ \forall \ell \in [m]$ (unfair) one, we can observe that the increase in α values are mainly accounted due to reassignments in round-robin procedure to satisfy group fairness. Notably, the distribution of `FAIRLOC` is better than group fair FRAC_{OE} and Jung+RR methods. All these plots are provided in Figure 4.7 for $\boldsymbol{\tau}_0$ and 4.8 for $\boldsymbol{\tau}_2$.

Runtime analysis

Next, we compare run-time on the US-West for opening ten NSS facilities. The average run-time over 10 different runs is reported in Table 4.2. While `FAIRLOC` uses the brute force technique to find the optimal facility for each agent, its runtime is similar to other

FAIRLOC	FRAC_{OE}	Jung	Jung+RR	LSPP
322.19	353.35	354.12	643.29	359.41

Table 4.2: Runtime comparison on US-West, NSS with k=10 at τ_2.

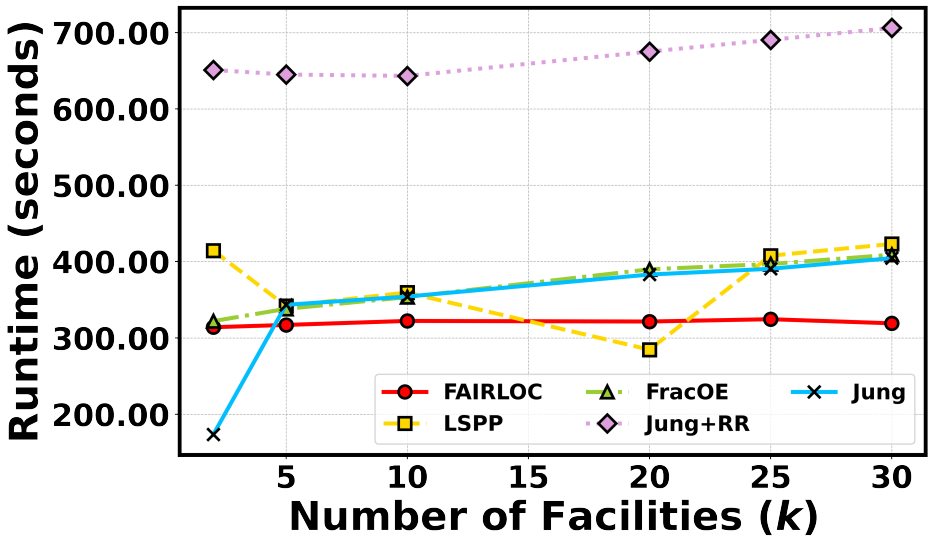


Figure 4.6: The plot shows runtime for different methods and FAIRLOC across varying k.

benchmarks. This is because other benchmarks either need tuning of parameters (Jung, Jung+RR, LSPP) or use local swaps (LSPP), or need to compute mean over all agent locations (FRAC_{OE}). The plot over varying k also shows that the runtime of FAIRLOC does not increase significantly with k and can be found in Figure 4.6.

4.7 Conclusion

The chapter proposes FAIRLOC that uses Nash social welfare to address the problem of satisfying group fairness in facility location problem. The method does not require hyper-parameter tuning, works for opening any number of facilities (k) and applies to any h-dimensional space. Further, the method allows for using an explicit set of possible facility opening locations. The algorithm has a quadratic approximation to the product objective (or two approximations to the logarithmic sum of distances) for k= 2 and conjecture for any k. We leave devising a novel proofing technique for quadratic approximation in general k. We further validate the method's efficacy on real-world United States census datasets and actual road maps for various settings. The findings showcase that FAIRLOC helps satisfy group fairness while achieving equitable (individually) fair assignment of agents. Another interesting future direction is to explore handling the strategic behaviour of agents and incentivizing them to behave rationally.

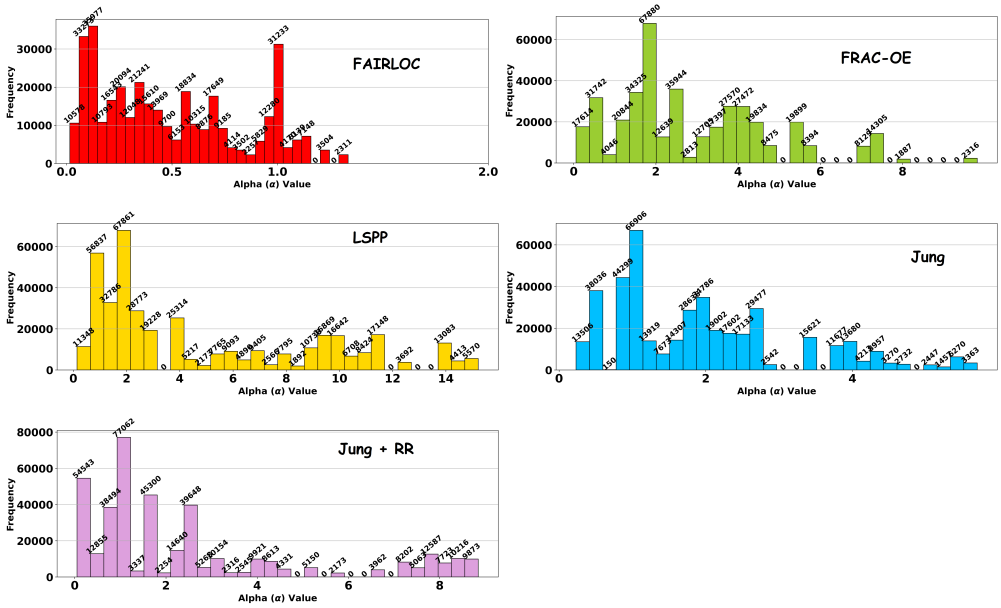


Figure 4.7: The plot shows distribution of α values for different methods on τ_0 in US-West for NSS at k=10.

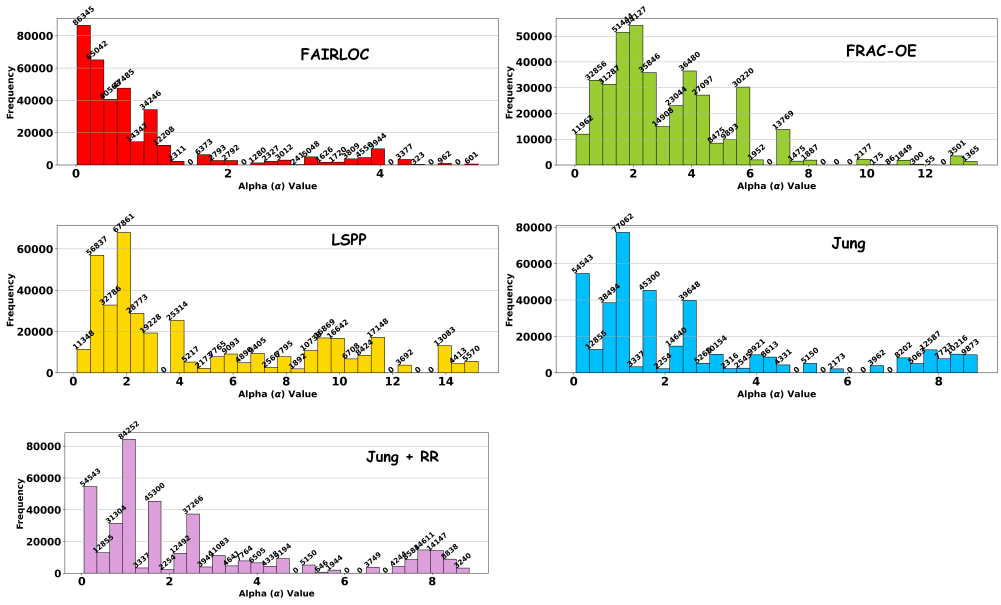


Figure 4.8: The plot shows distribution of α values for different methods on τ_2 in US-West for NSS at k=10.

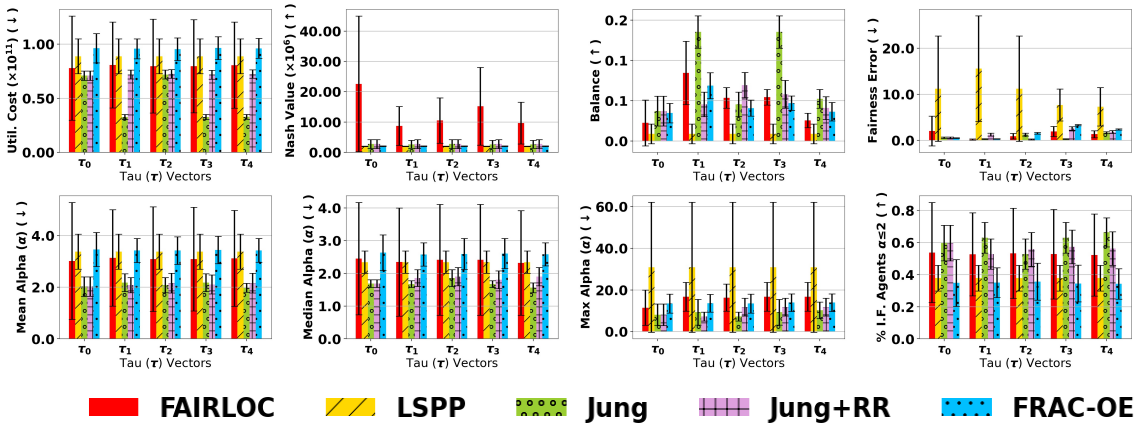


Figure 4.9: The plot shows the variation in metrics across different τ vectors for opening **Dialysis Centers** in the **US-Midwest** region, with **income level** as the protected group. The first row displays methods for utilitarian cost, Nash value, group fairness metrics balance and fairness error. The second row compares the α values distribution using mean, median, max and the number of agents having $\alpha \leq 2$. The arrow on the Y-axis indicates a favorable direction.

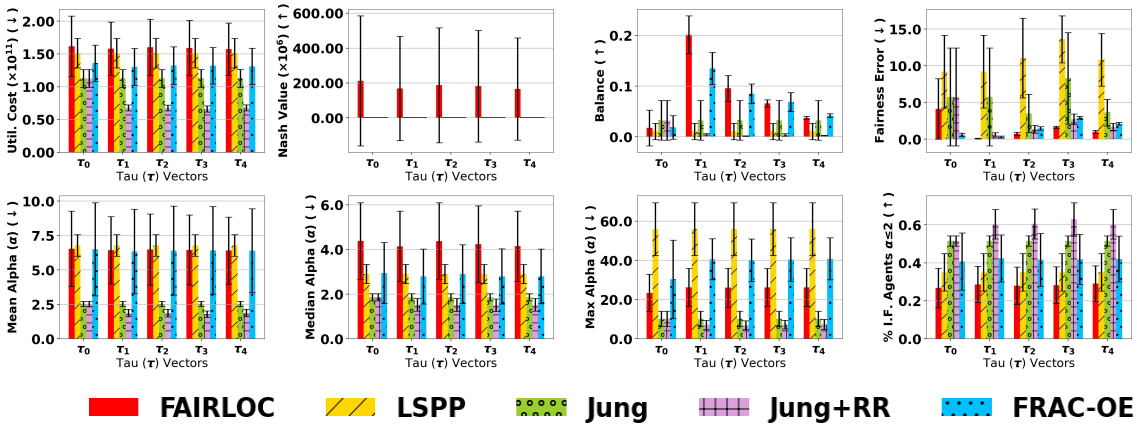


Figure 4.10: The plot shows the variation in metrics across different τ vectors for opening **Dialysis Centers** in the **US-West** region, with **income level** as the protected group. The first row displays methods for utilitarian cost, Nash value, group fairness metrics balance and fairness error. The second row compares the α values distribution using mean, median, max and the number of agents having $\alpha \leq 2$. The arrow on the Y-axis indicates a favorable direction.

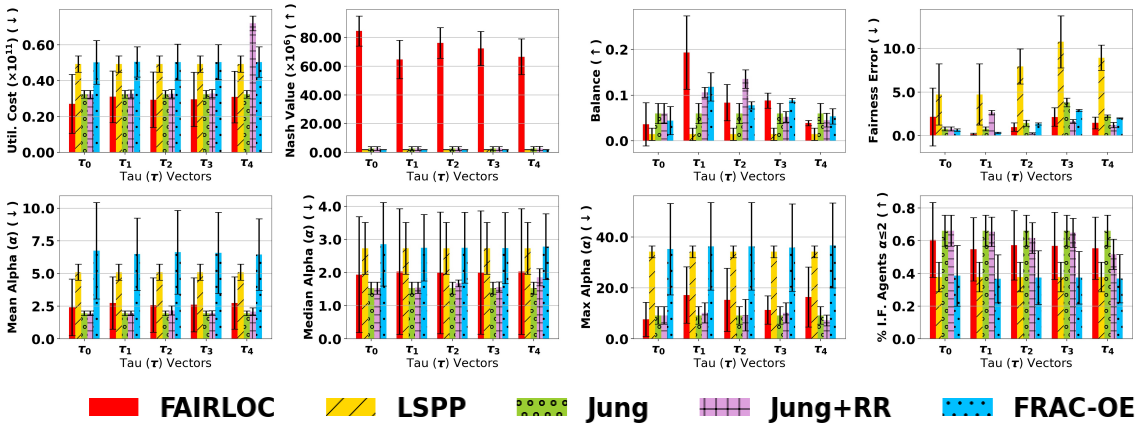


Figure 4.11: The plot shows the variation in metrics across different τ vectors for opening **Dialysis Centers** in the **US-North** region, with **income level** as the protected group. The first row displays methods for utilitarian cost, Nash value, group fairness metrics balance and fairness error. The second row compares the α values distribution using mean, median, max and the number of agents having $\alpha \leq 2$. The arrow on the Y-axis indicates a favorable direction.

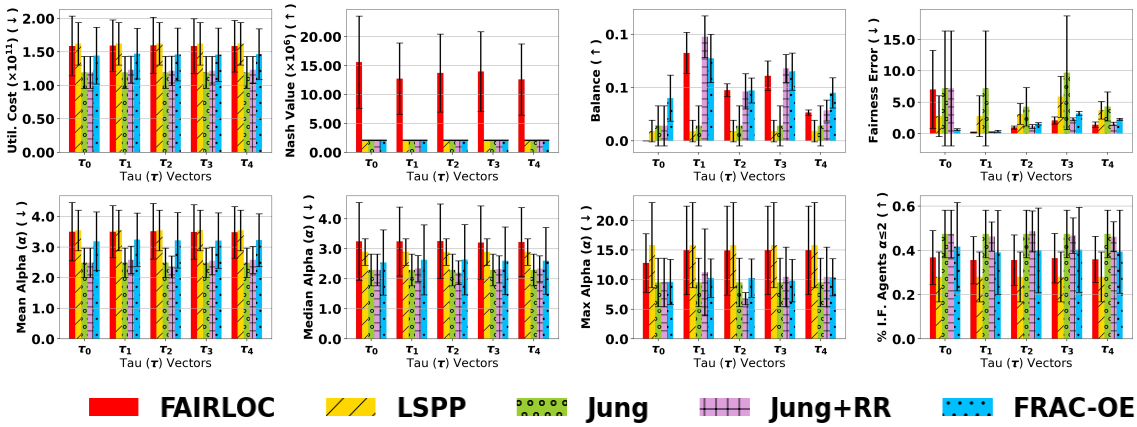


Figure 4.12: The plot shows the variation in metrics across different τ vectors for opening **Dialysis Centers** in the **US-South** region, with **income level** as the protected group. The first row displays methods for utilitarian cost, Nash value, group fairness metrics balance and fairness error. The second row compares the α values distribution using mean, median, max and the number of agents having $\alpha \leq 2$. The arrow on the Y-axis indicates a favorable direction.

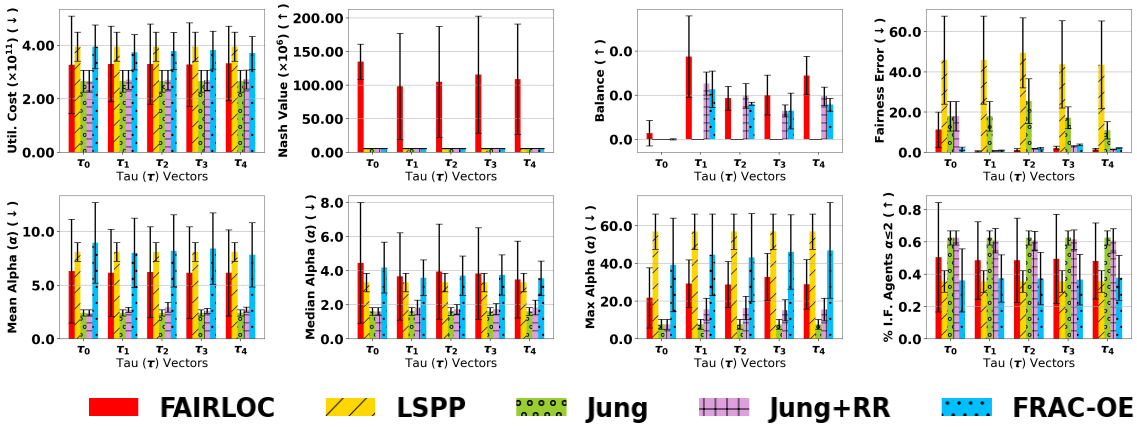


Figure 4.13: The plot shows the variation in metrics across different τ vectors for opening **Dialysis Centers** in the **US-Midwest** region, with **race** as the protected group. The first row displays methods for utilitarian cost, Nash value, group fairness metrics balance and fairness error. The second row compares the α values distribution using mean, median, max and the number of agents having $\alpha \leq 2$. The arrow on the Y-axis indicates a favorable direction.

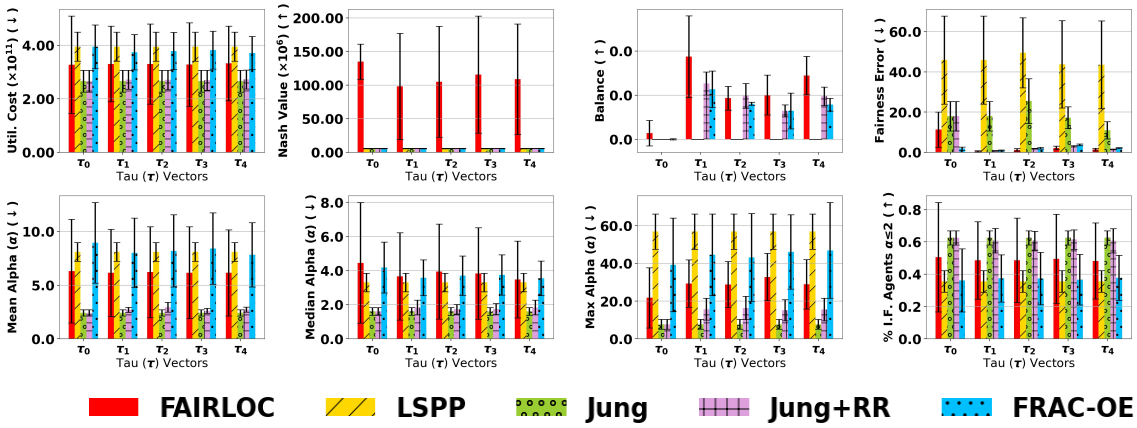


Figure 4.14: The plot shows the variation in metrics across different τ vectors for opening **Dialysis Centers** in the **US-West** region, with **race** as the protected group. The first row displays methods for utilitarian cost, Nash value, group fairness metrics balance and fairness error. The second row compares the α values distribution using mean, median, max and the number of agents having $\alpha \leq 2$. The arrow on the Y-axis indicates a favorable direction.

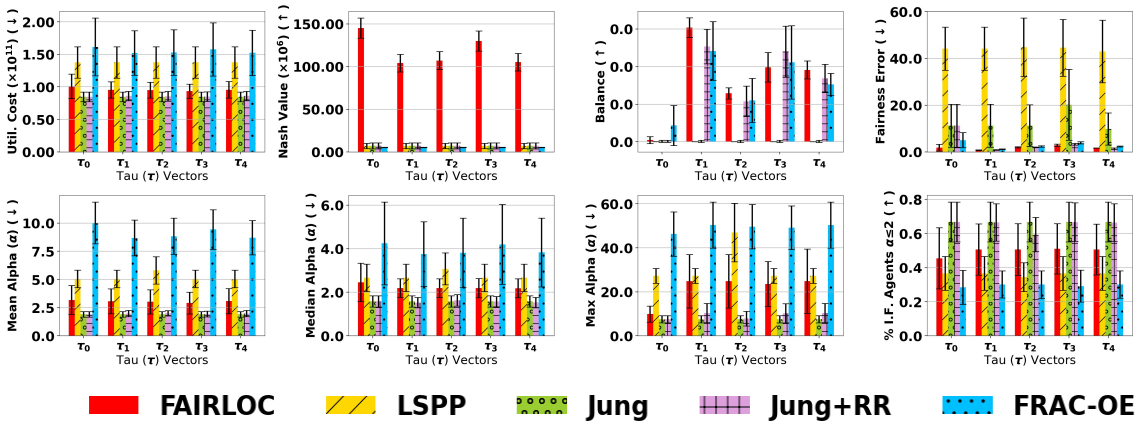


Figure 4.15: The plot shows the variation in metrics across different τ vectors for opening **Dialysis Centers** in the **US-North** region, with **race** as the protected group. The first row displays methods for utilitarian cost, Nash value, group fairness metrics balance and fairness error. The second row compares the α values distribution using mean, median, max and the number of agents having $\alpha \leq 2$. The arrow on the Y-axis indicates a favorable direction.

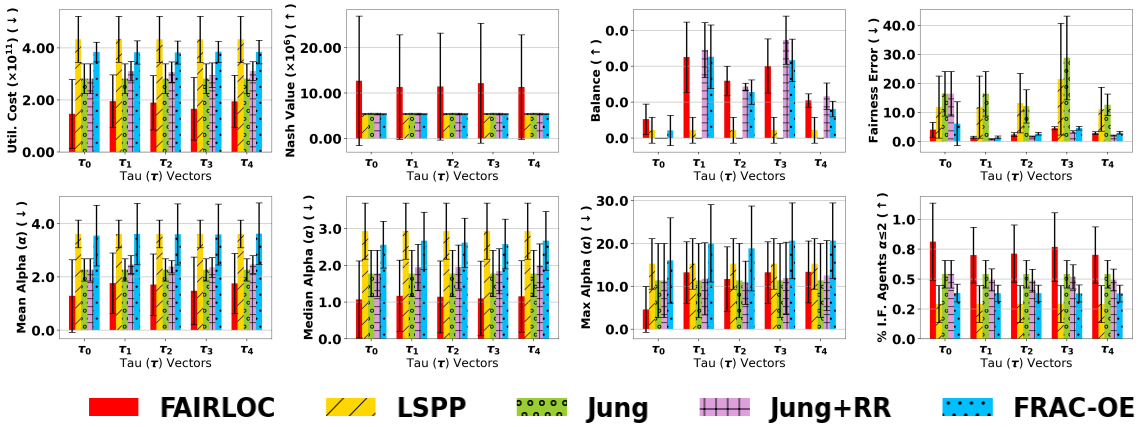


Figure 4.16: The plot shows the variation in metrics across different τ vectors for opening **Dialysis Centers** in the **US-South** region, with **race** as the protected group. The first row displays methods for utilitarian cost, Nash value, group fairness metrics balance and fairness error. The second row compares the α values distribution using mean, median, max and the number of agents having $\alpha \leq 2$. The arrow on the Y-axis indicates a favorable direction.

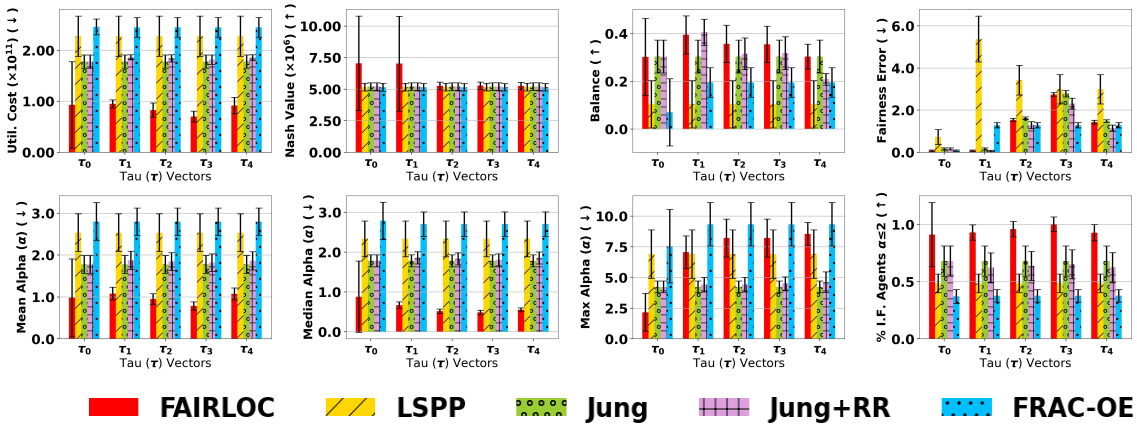


Figure 4.17: The plot shows the variation in metrics across different τ vectors for opening **National Shelter Systems (NSS)** in the **US-Midwest** region, with **age-gender** as the protected group. The first row displays methods for utilitarian cost, Nash value, group fairness metrics balance and fairness error. The second row compares the α values distribution using mean, median, max and the number of agents having $\alpha \leq 2$. The arrow on the Y-axis indicates a favorable direction.

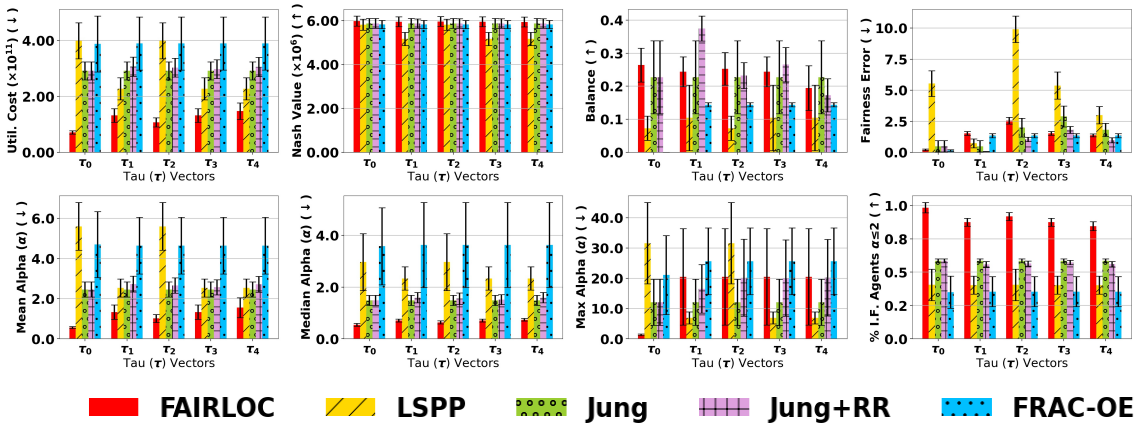


Figure 4.18: The plot shows the variation in metrics across different τ vectors for opening **National Shelter Systems (NSS)** in the **US-West** region, with **age-gender** as the protected group. The first row displays methods for utilitarian cost, Nash value, group fairness metrics balance and fairness error. The second row compares the α values distribution using mean, median, max and the number of agents having $\alpha \leq 2$. The arrow on the Y-axis indicates a favorable direction.

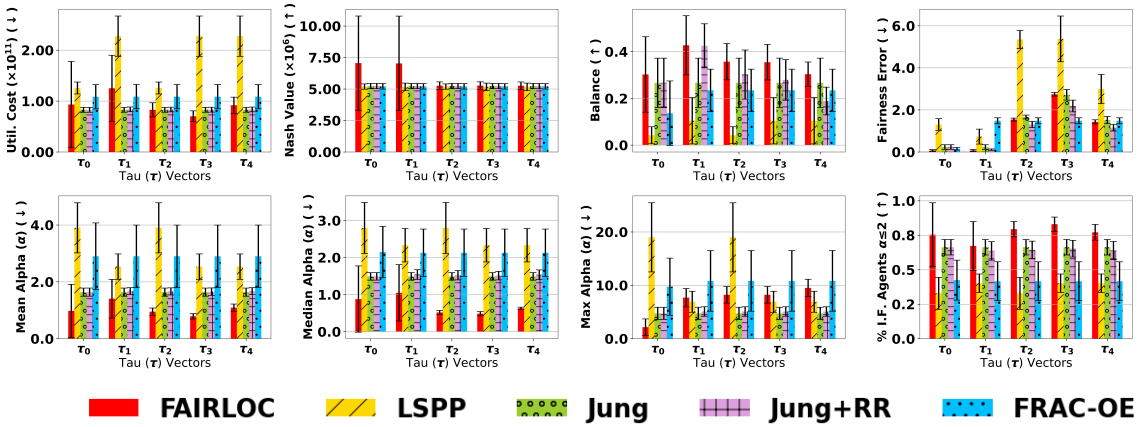


Figure 4.19: The plot shows the variation in metrics across different τ vectors for opening **National Shelter Systems (NSS)** in the **US-North** region, with **age-gender** as the protected group. The first row displays methods for utilitarian cost, Nash value, group fairness metrics balance and fairness error. The second row compares the α values distribution using mean, median, max and the number of agents having $\alpha \leq 2$. The arrow on the Y-axis indicates a favorable direction.

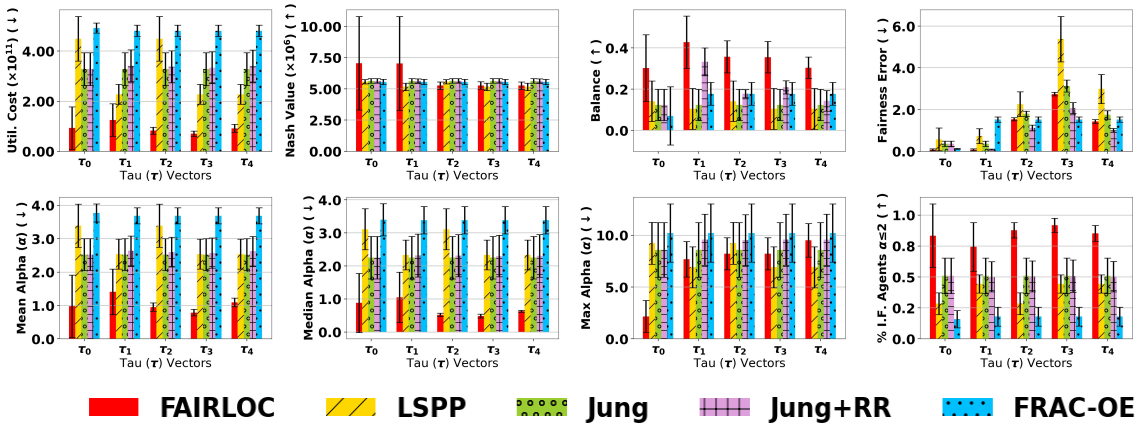


Figure 4.20: The plot shows the variation in metrics across different τ vectors for opening **National Shelter Systems (NSS)** in the **US-South** region, with **age-gender** as the protected group. The first row displays methods for utilitarian cost, Nash value, group fairness metrics balance and fairness error. The second row compares the α values distribution using mean, median, max and the number of agents having $\alpha \leq 2$. The arrow on the Y-axis indicates a favorable direction.

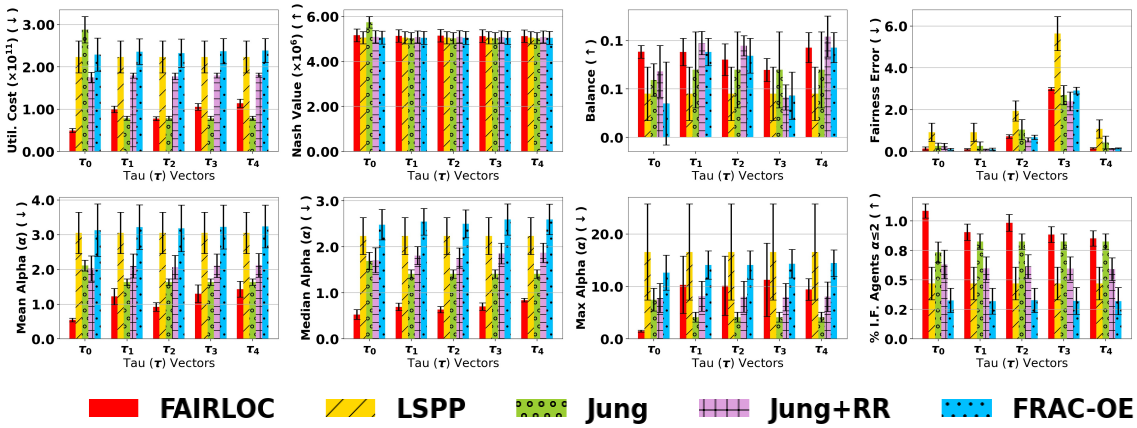


Figure 4.21: The plot shows the variation in metrics across different τ vectors for opening **Pharmacy** in the **US-Midwest** region, with **poverty** level as the protected group. The first row displays methods for utilitarian cost, Nash value, group fairness metrics balance and fairness error. The second row compares the α values distribution using mean, median, max and the number of agents having $\alpha \leq 2$. The arrow on the Y-axis indicates a favorable direction.

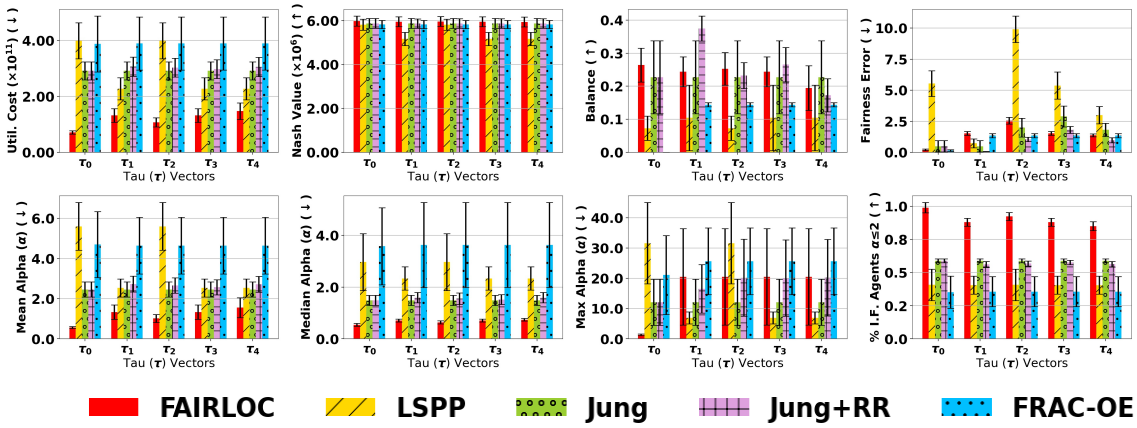


Figure 4.22: The plot shows the variation in metrics across different τ vectors for opening **Pharmacy** in the **US-West** region, with **poverty** level as the protected group. The first row displays methods for utilitarian cost, Nash value, group fairness metrics balance and fairness error. The second row compares the α values distribution using mean, median, max and the number of agents having $\alpha \leq 2$. The arrow on the Y-axis indicates a favorable direction.

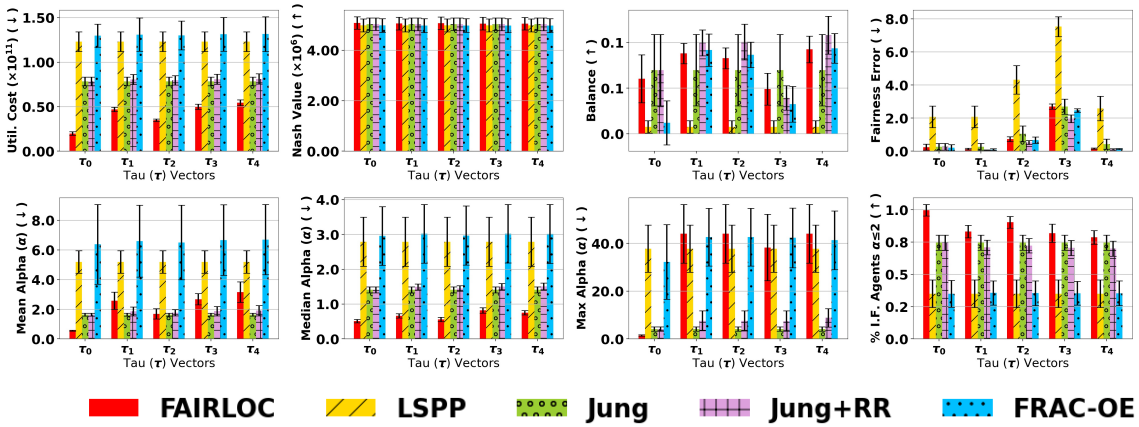


Figure 4.23: The plot shows the variation in metrics across different τ vectors for opening **Pharmacy** in the **US-North** region, with **poverty** level as the protected group. The first row displays methods for utilitarian cost, Nash value, group fairness metrics balance and fairness error. The second row compares the α values distribution using mean, median, max and the number of agents having $\alpha \leq 2$. The arrow on the Y-axis indicates a favorable direction.

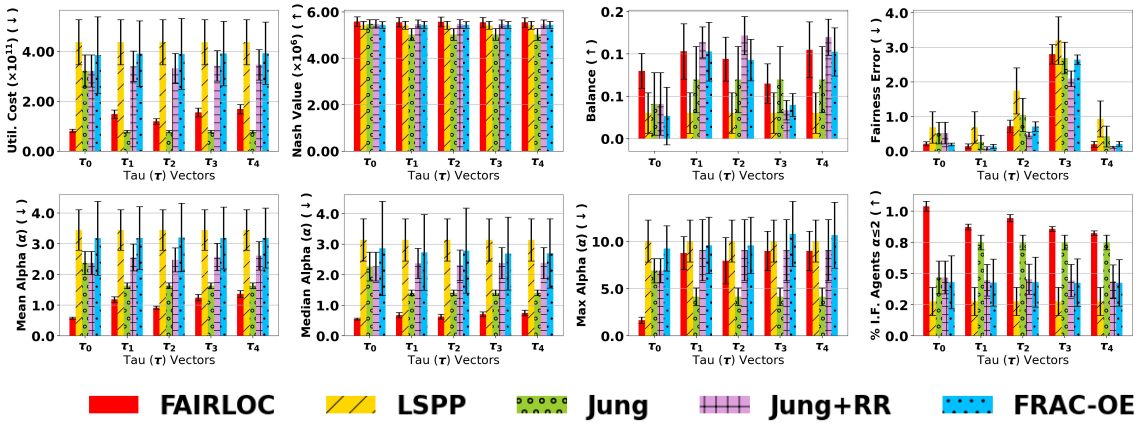


Figure 4.24: The plot shows the variation in metrics across different τ vectors for opening **Pharmacy** in the **US-South** region, with **poverty** level as the protected group. The first row displays methods for utilitarian cost, Nash value, group fairness metrics balance and fairness error. The second row compares the α values distribution using mean, median, max and the number of agents having $\alpha \leq 2$. The arrow on the Y-axis indicates a favorable direction.

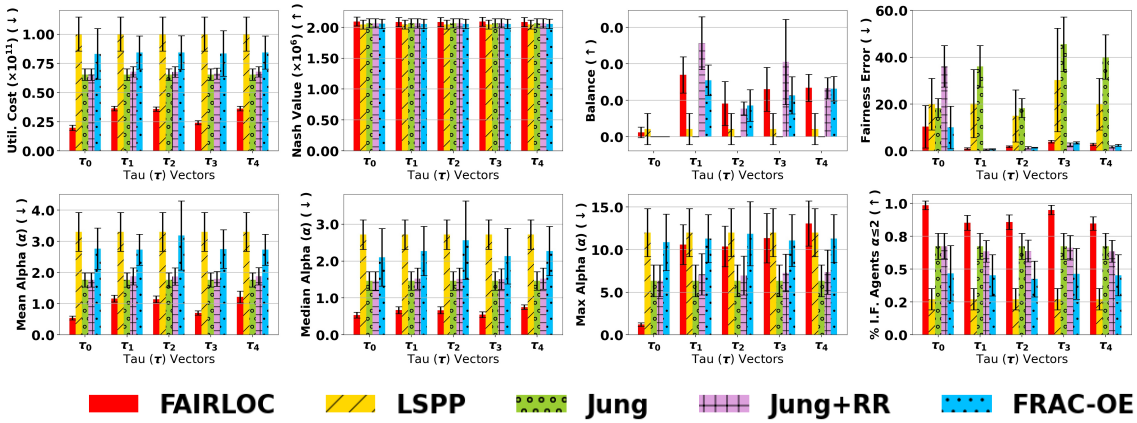


Figure 4.25: The plot shows the variation in metrics across different τ vectors for opening **Schools** in the **US-Midwest** region, with **language** as the protected group. The first row displays methods for utilitarian cost, Nash value, group fairness metrics balance and fairness error. The second row compares the α values distribution using mean, median, max and the number of agents having $\alpha \leq 2$. The arrow on the Y-axis indicates a favorable direction.

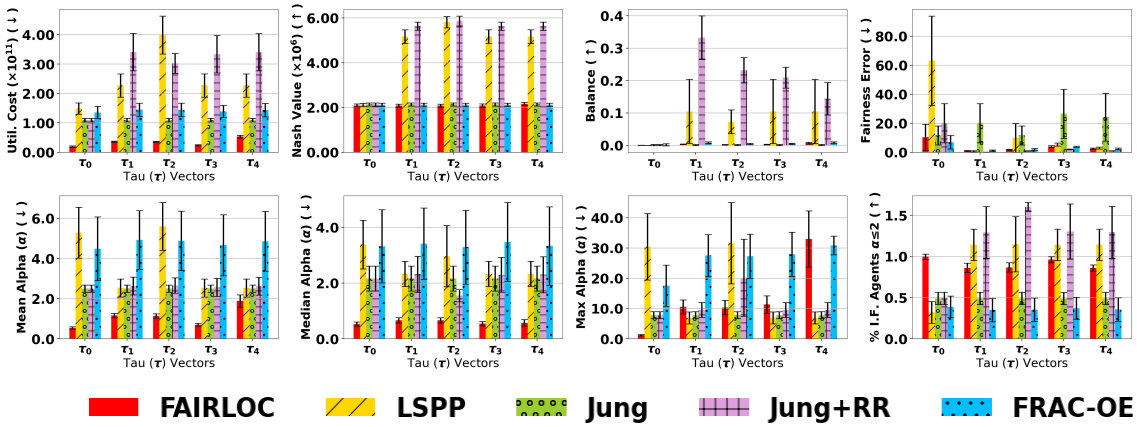


Figure 4.26: The plot shows the variation in metrics across different τ vectors for opening **Schools** in the **US-West** region, with **language** as the protected group. The first row displays methods for utilitarian cost, Nash value, group fairness metrics balance and fairness error. The second row compares the α values distribution using mean, median, max and the number of agents having $\alpha \leq 2$. The arrow on the Y-axis indicates a favorable direction.

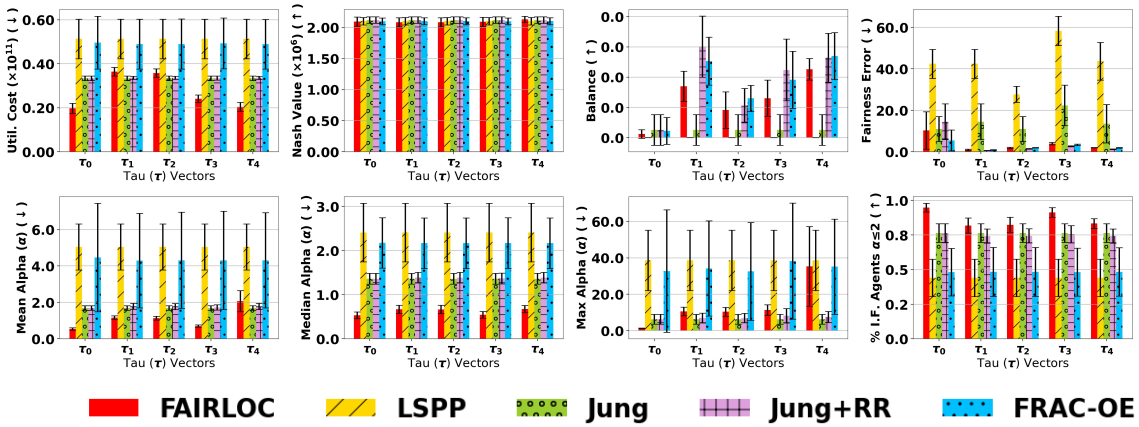


Figure 4.27: The plot shows the variation in metrics across different τ vectors for opening **Schools** in the **US-North** region, with **language** as the protected group. The first row displays methods for utilitarian cost, Nash value, group fairness metrics balance and fairness error. The second row compares the α values distribution using mean, median, max and the number of agents having $\alpha \leq 2$. The arrow on the Y-axis indicates a favorable direction.

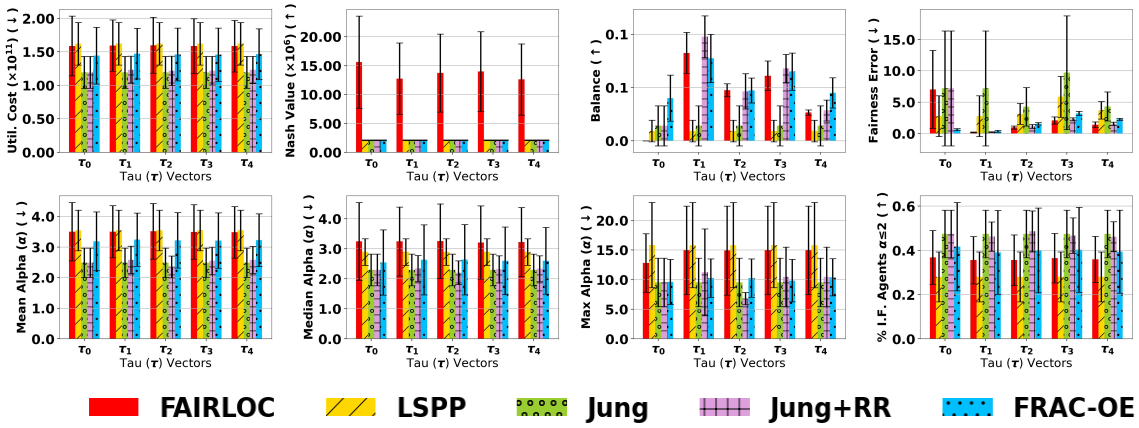


Figure 4.28: The plot shows the variation in metrics across different τ vectors for opening **Schools** in the **US-South** region, with **language** as the protected group. The first row displays methods for utilitarian cost, Nash value, group fairness metrics balance and fairness error. The second row compares the α values distribution using mean, median, max and the number of agents having $\alpha \leq 2$. The arrow on the Y-axis indicates a favorable direction.

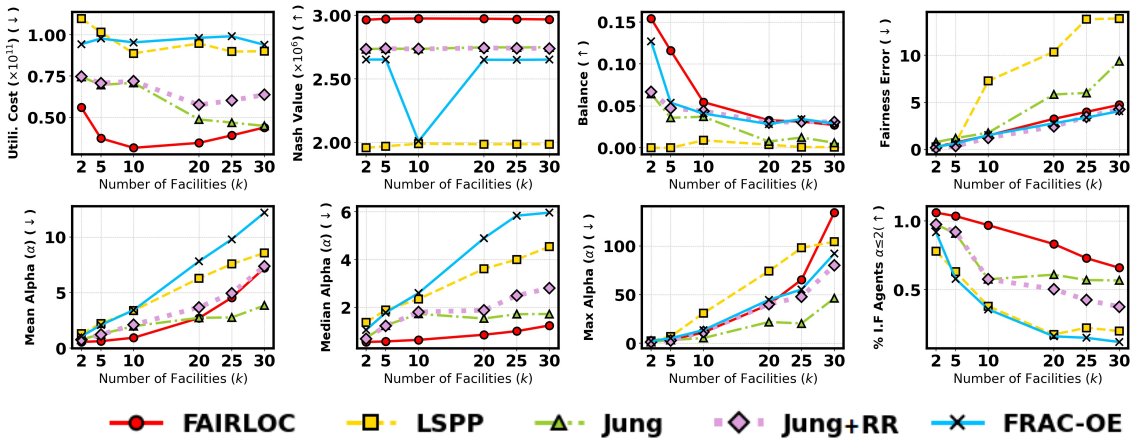


Figure 4.29: The plot shows the variation in metrics across varying k for opening **Dialysis Centers** in the **US-Midwest** region, with **income level** as the protected group. The first row displays methods for utilitarian cost, Nash value, group fairness metrics balance and fairness error. The second row compares the α values distribution using mean, median, max and the number of agents having $\alpha \leq 2$. The arrow on the Y-axis indicates a favorable direction.

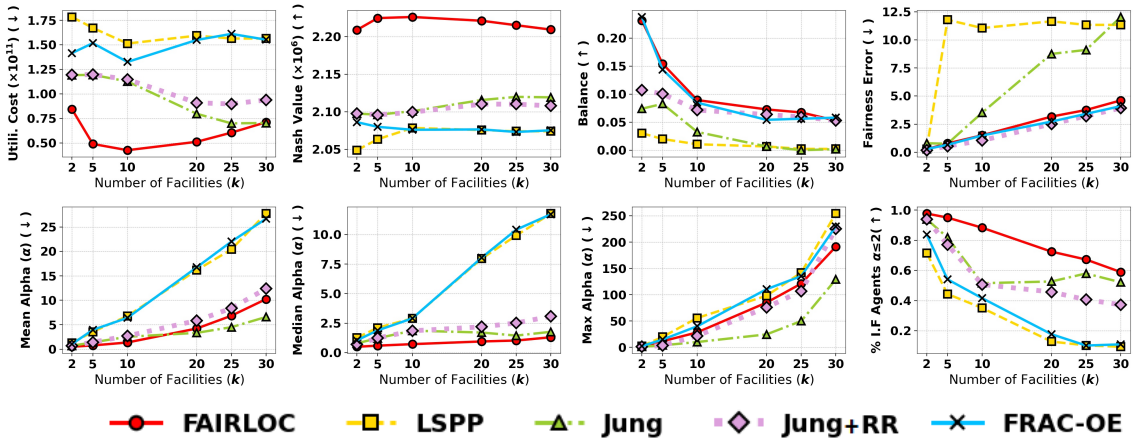


Figure 4.30: The plot shows the variation in metrics across varying k for opening **Dialysis Centers** in the **US-West** region, with **income level** as the protected group. The first row displays methods for utilitarian cost, Nash value, group fairness metrics balance and fairness error. The second row compares the α values distribution using mean, median, max and the number of agents having $\alpha \leq 2$. The arrow on the Y-axis indicates a favorable direction.

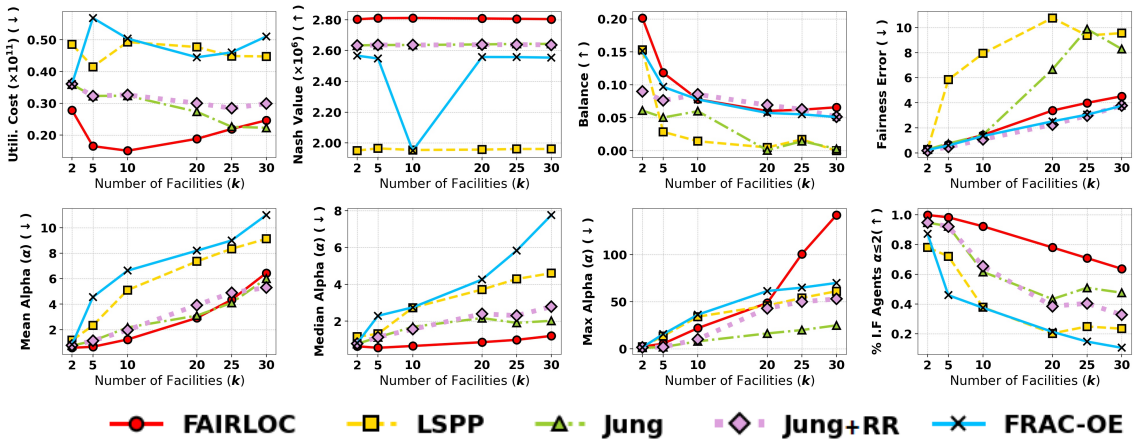


Figure 4.31: The plot shows the variation in metrics across varying k for opening **Dialysis Centers** in the **US-North** region, with **income level** as the protected group. The first row displays methods for utilitarian cost, Nash value, group fairness metrics balance and fairness error. The second row compares the α values distribution using mean, median, max and the number of agents having $\alpha \leq 2$. The arrow on the Y-axis indicates a favorable direction.

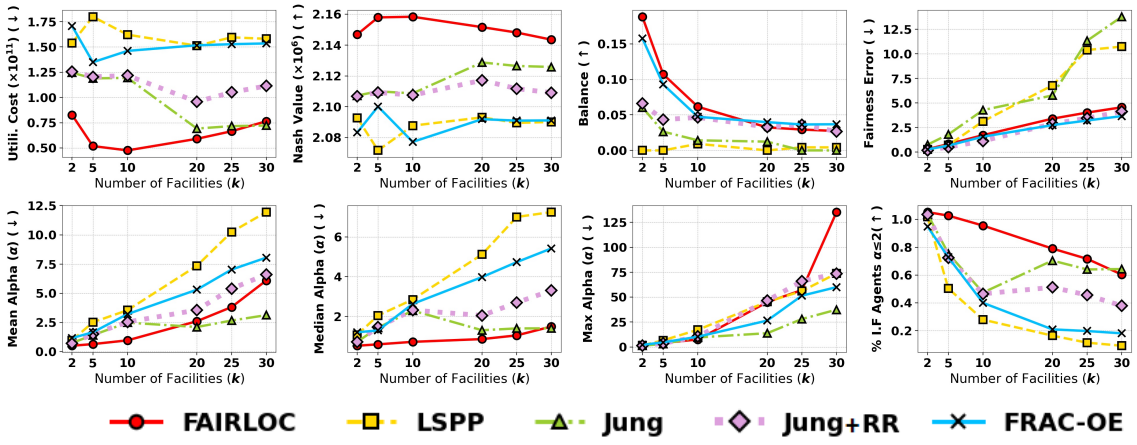


Figure 4.32: The plot shows the variation in metrics across varying k for opening **Dialysis Centers** in the **US-South** region, with **income level** as the protected group. The first row displays methods for utilitarian cost, Nash value, group fairness metrics balance and fairness error. The second row compares the α values distribution using mean, median, max and the number of agents having $\alpha \leq 2$. The arrow on the Y-axis indicates a favorable direction.

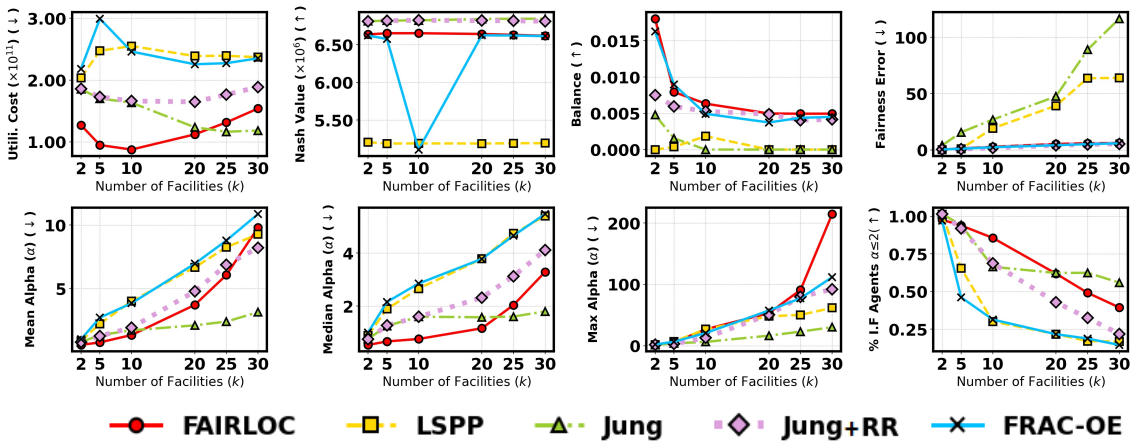


Figure 4.33: The plot shows the variation in metrics across varying k for opening **Dialysis Centers** in the **US-Midwest** region, with **race** as the protected group. The first row displays methods for utilitarian cost, Nash value, group fairness metrics balance and fairness error. The second row compares the α values distribution using mean, median, max and the number of agents having $\alpha \leq 2$. The arrow on the Y-axis indicates a favorable direction.

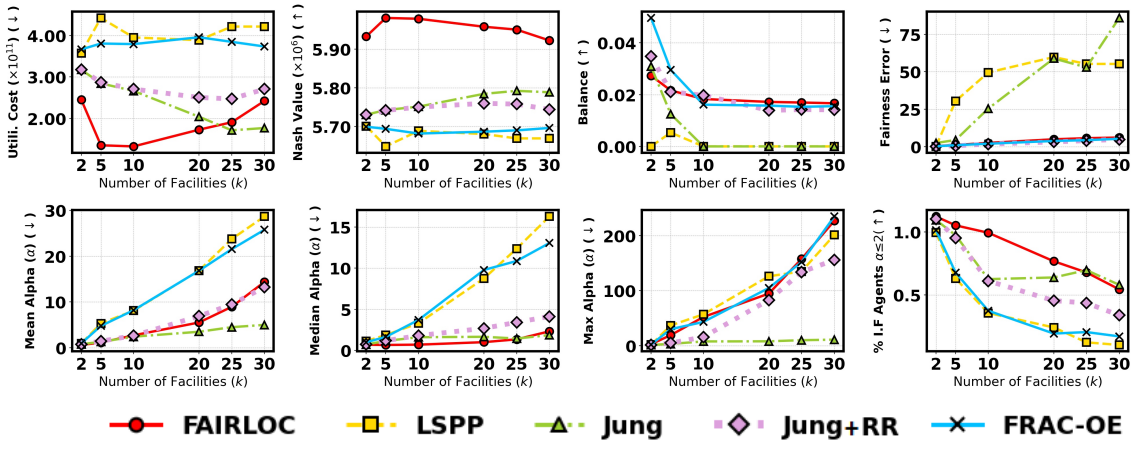


Figure 4.34: The plot shows the variation in metrics across varying k for opening **Dialysis Race** in the **US-West** region, with **race** as the protected group. The first row displays methods for utilitarian cost, Nash value, group fairness metrics balance and fairness error. The second row compares the α values distribution using mean, median, max and the number of agents having $\alpha \leq 2$. The arrow on the Y-axis indicates a favorable direction.

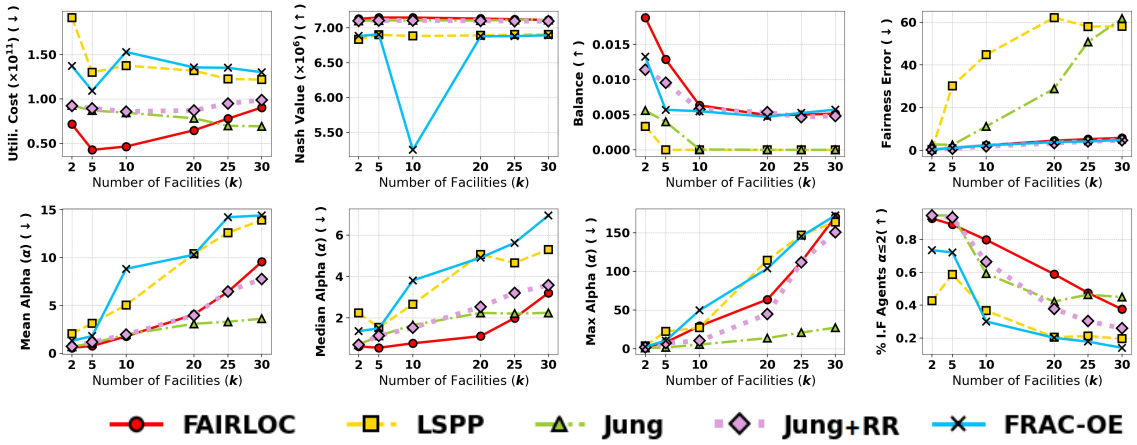


Figure 4.35: The plot shows the variation in metrics across varying k for opening **Dialysis Race** in the **US-North** region, with **race** as the protected group. The first row displays methods for utilitarian cost, Nash value, group fairness metrics balance and fairness error. The second row compares the α values distribution using mean, median, max and the number of agents having $\alpha \leq 2$. The arrow on the Y-axis indicates a favorable direction.

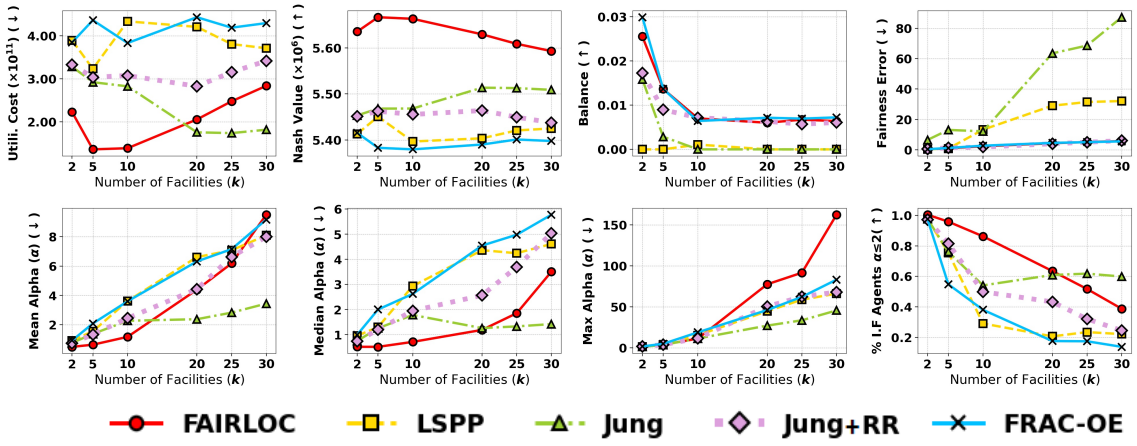


Figure 4.36: The plot shows the variation in metrics across varying k for opening **Dialysis Race** in the **US-South** region, with **race** as the protected group. The first row displays methods for utilitarian cost, Nash value, group fairness metrics balance and fairness error. The second row compares the α values distribution using mean, median, max and the number of agents having $\alpha \leq 2$. The arrow on the Y-axis indicates a favorable direction.

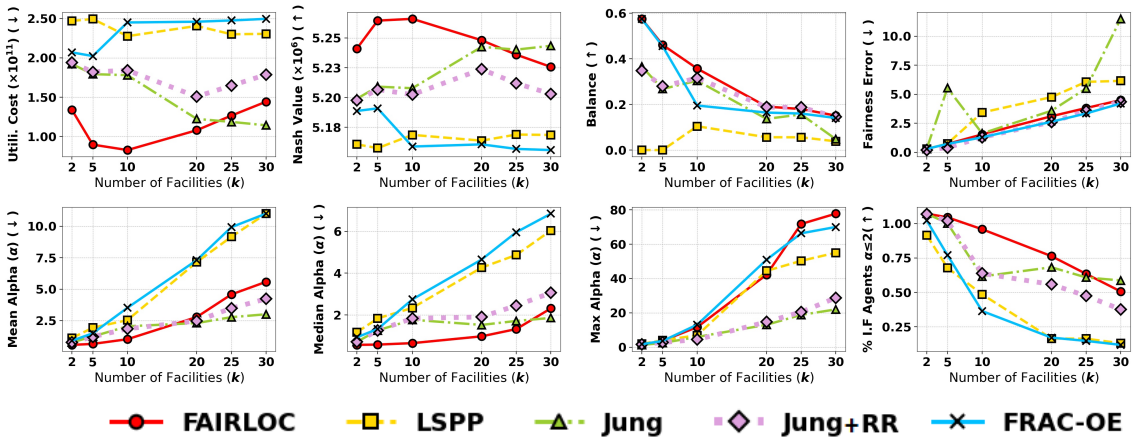


Figure 4.37: The plot shows the variation in metrics across varying k for opening **National Shelter Systems (NSS)** in the **US-Midwest** region, with **age gender** as the protected group. The first row displays methods for utilitarian cost, Nash value, group fairness metrics balance and fairness error. The second row compares the α values distribution using mean, median, max and the number of agents having $\alpha \leq 2$. The arrow on the Y-axis indicates a favorable direction.

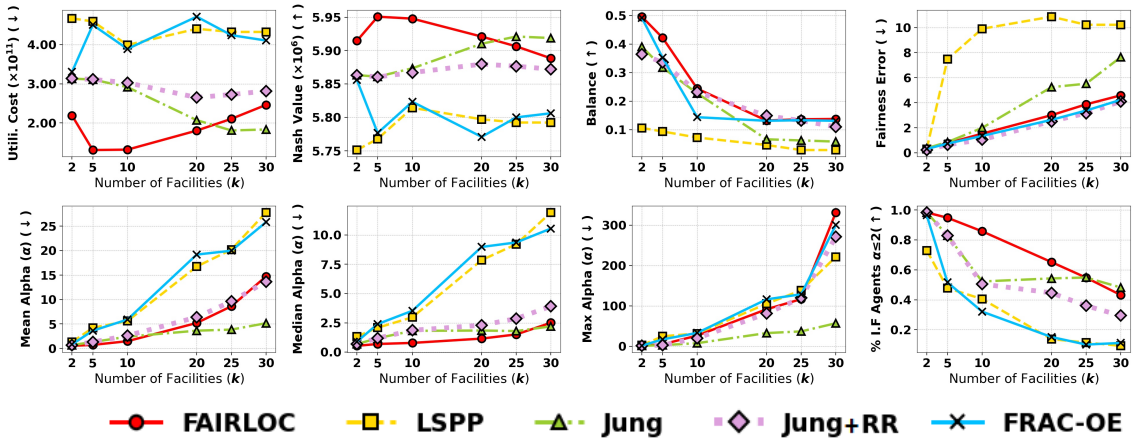


Figure 4.38: The plot shows the variation in metrics across varying k for opening **National Shelter Systems (NSS)** in the **US-West** region, with **age gender** as the protected group. The first row displays methods for utilitarian cost, Nash value, group fairness metrics balance and fairness error. The second row compares the α values distribution using mean, median, max and the number of agents having $\alpha \leq 2$. The arrow on the Y-axis indicates a favorable direction.

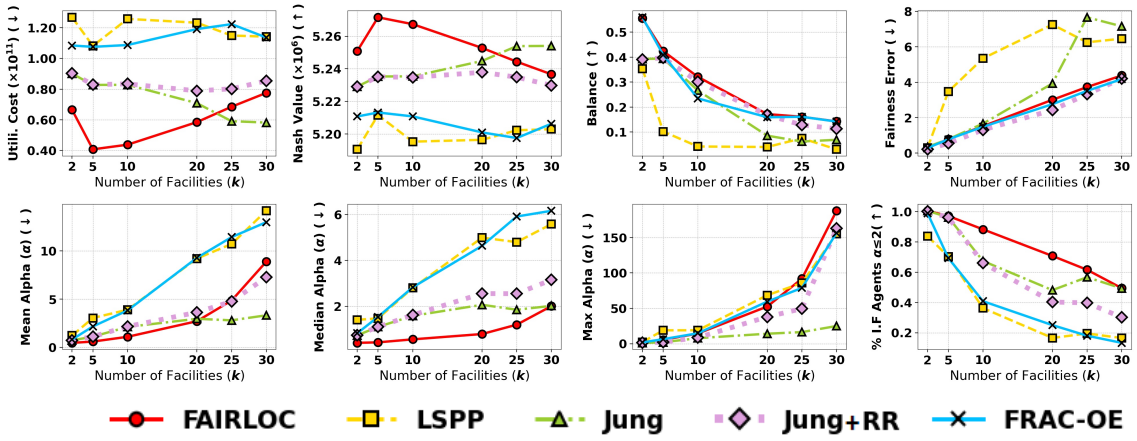


Figure 4.39: The plot shows the variation in metrics across varying k for opening **National Shelter Systems (NSS)** in the **US-North** region, with **age gender** as the protected group. The first row displays methods for utilitarian cost, Nash value, group fairness metrics balance and fairness error. The second row compares the α values distribution using mean, median, max and the number of agents having $\alpha \leq 2$. The arrow on the Y-axis indicates a favorable direction.

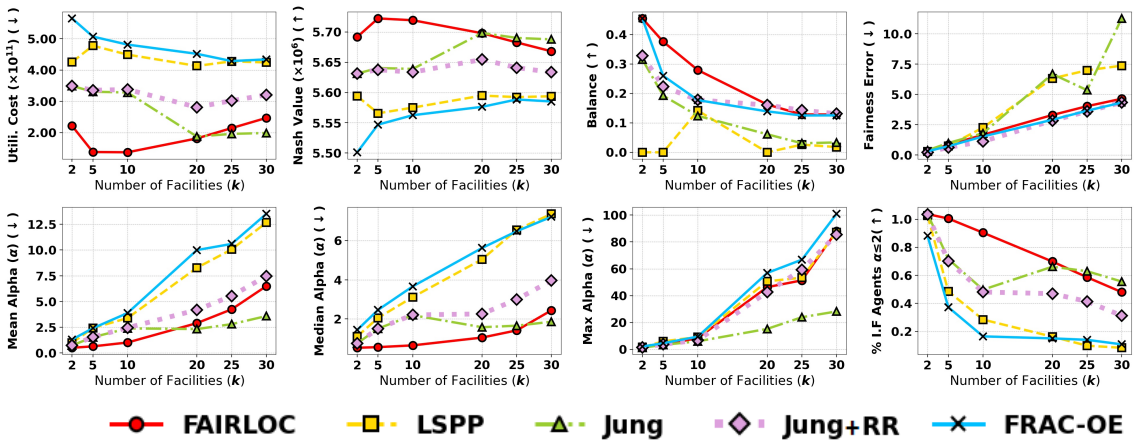


Figure 4.40: The plot shows the variation in metrics across varying k for opening **National Shelter Systems (NSS)** in the **US-South** region, with **age gender** as the protected group. The first row displays methods for utilitarian cost, Nash value, group fairness metrics balance and fairness error. The second row compares the α values distribution using mean, median, max and the number of agents having $\alpha \leq 2$. The arrow on the Y-axis indicates a favorable direction.

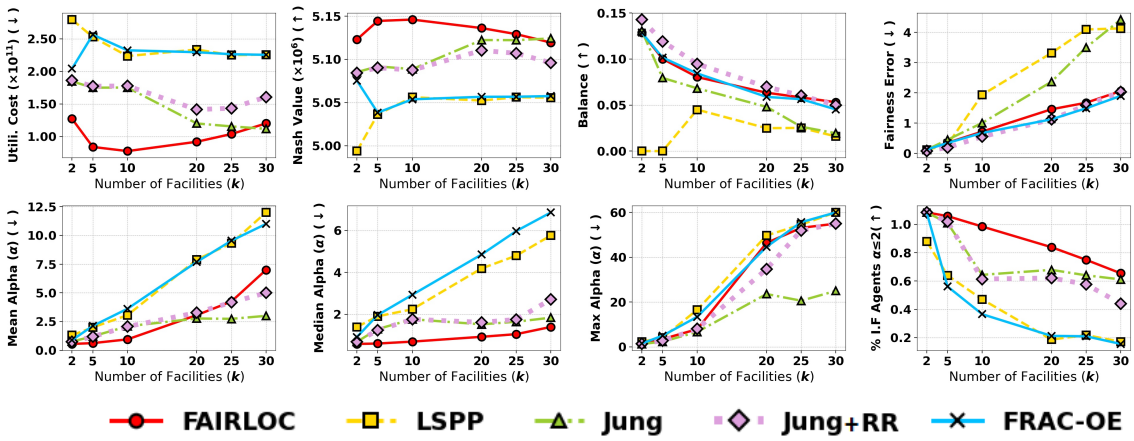


Figure 4.41: The plot shows the variation in metrics across varying k for opening **Pharmacy** in the **US-Midwest** region, with **poverty** levels as the protected group. The first row displays methods for utilitarian cost, Nash value, group fairness metrics balance and fairness error. The second row compares the α values distribution using mean, median, max and the number of agents having $\alpha \leq 2$. The arrow on the Y-axis indicates a favorable direction.

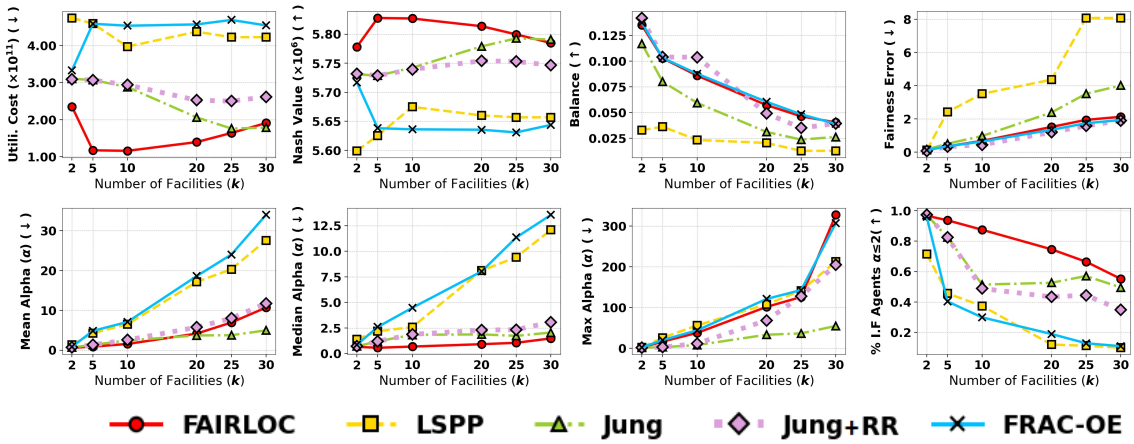


Figure 4.42: The plot shows the variation in metrics across varying k for opening **Pharmacy** in the **US-West** region, with **poverty** levels as the protected group. The first row displays methods for utilitarian cost, Nash value, group fairness metrics balance and fairness error. The second row compares the α values distribution using mean, median, max and the number of agents having $\alpha \leq 2$. The arrow on the Y-axis indicates a favorable direction.

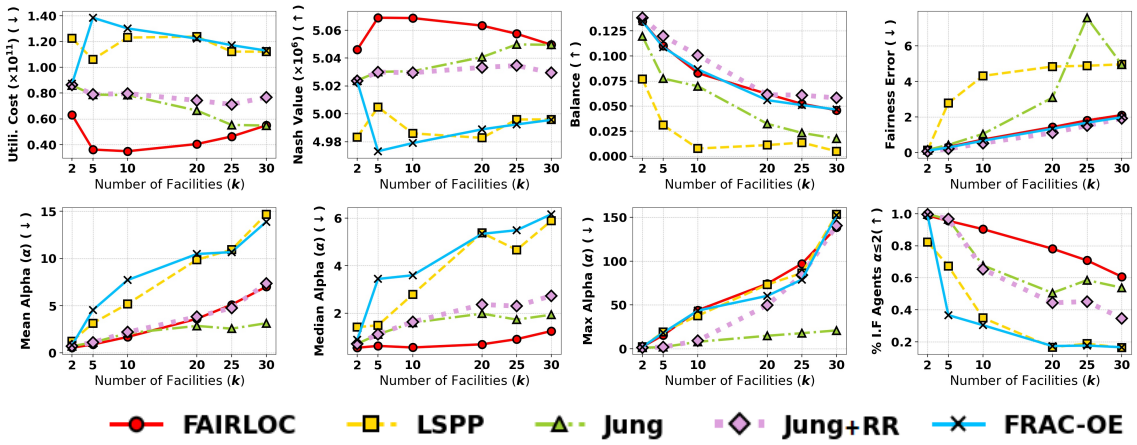


Figure 4.43: The plot shows the variation in metrics across varying k for opening **Pharmacy** in the **US-North** region, with **poverty** levels as the protected group. The first row displays methods for utilitarian cost, Nash value, group fairness metrics balance and fairness error. The second row compares the α values distribution using mean, median, max and the number of agents having $\alpha \leq 2$. The arrow on the Y-axis indicates a favorable direction.

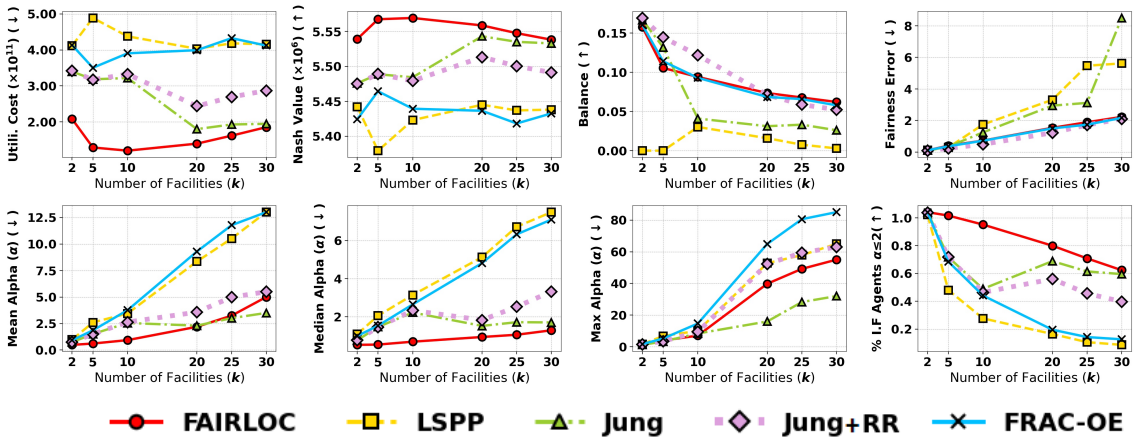


Figure 4.44: The plot shows the variation in metrics across varying k for opening **Pharmacy** in the **US-South** region, with **poverty** levels as the protected group. The first row displays methods for utilitarian cost, Nash value, group fairness metrics balance and fairness error. The second row compares the α values distribution using mean, median, max and the number of agents having $\alpha \leq 2$. The arrow on the Y-axis indicates a favorable direction.

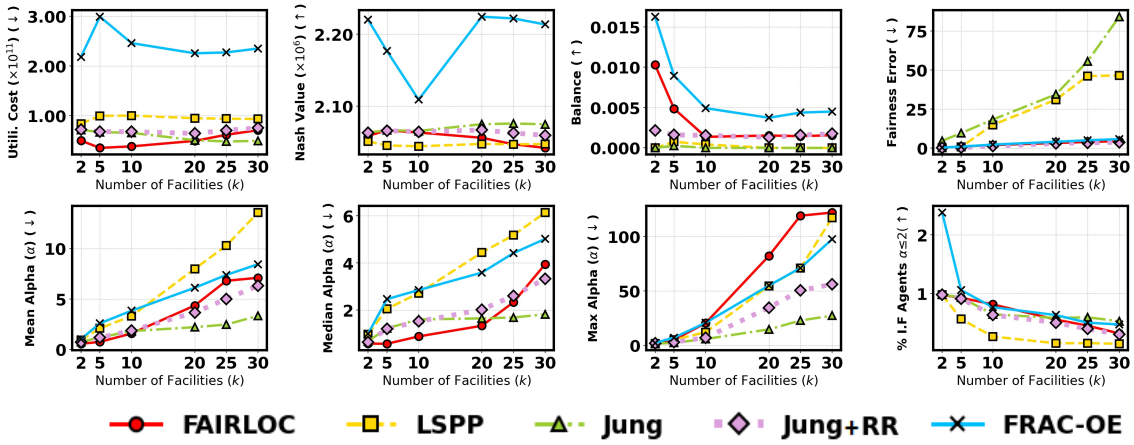


Figure 4.45: The plot shows the variation in metrics across varying k for opening **Schools** in the **US-Midwest** region, with **language** as the protected group. The first row displays methods for utilitarian cost, Nash value, group fairness metrics balance and fairness error. The second row compares the α values distribution using mean, median, max and the number of agents having $\alpha \leq 2$. The arrow on the Y-axis indicates a favorable direction.

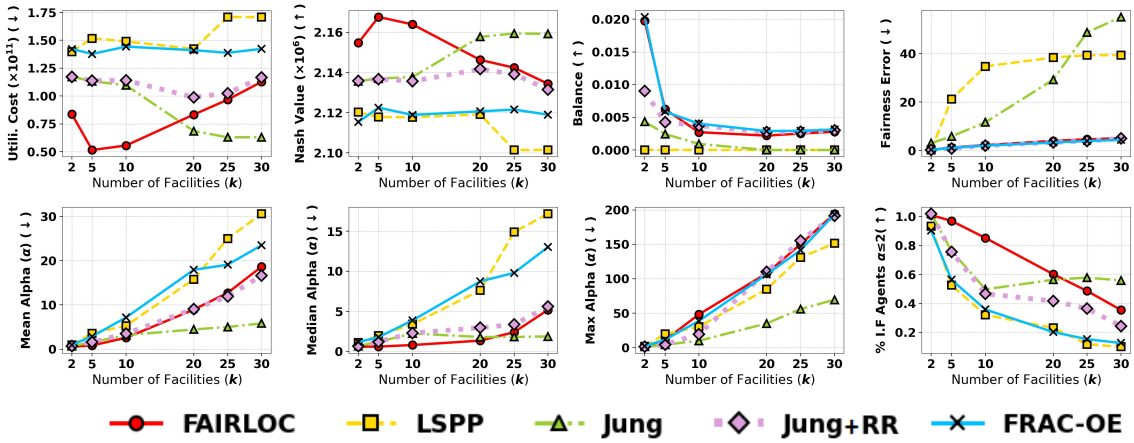


Figure 4.46: The plot shows the variation in metrics across varying k for opening **Schools** in the **US-West** region, with **language** as the protected group. The first row displays methods for utilitarian cost, Nash value, group fairness metrics balance and fairness error. The second row compares the α values distribution using mean, median, max and the number of agents having $\alpha \leq 2$. The arrow on the Y-axis indicates a favorable direction.

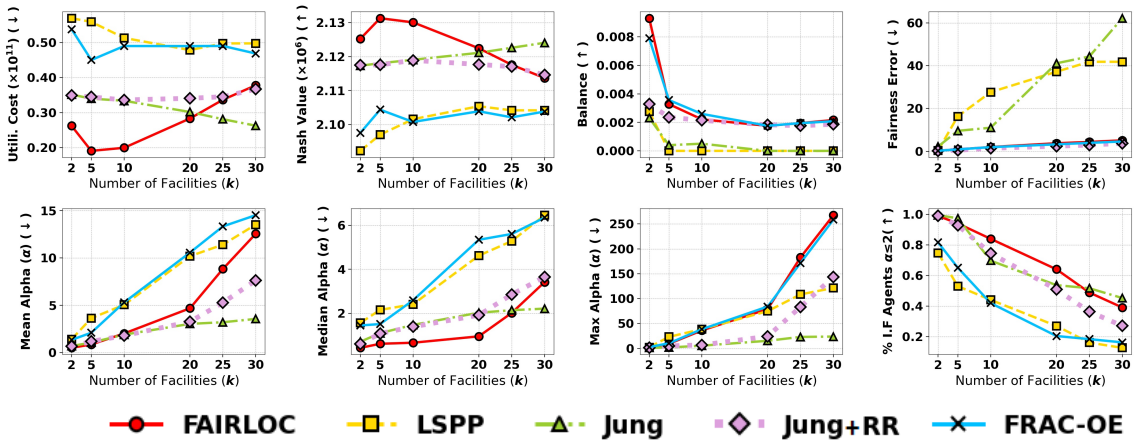


Figure 4.47: The plot shows the variation in metrics across varying k for opening **Schools** in the **US-North** region, with **language** as the protected group. The first row displays methods for utilitarian cost, Nash value, group fairness metrics balance and fairness error. The second row compares the α values distribution using mean, median, max and the number of agents having $\alpha \leq 2$. The arrow on the Y-axis indicates a favorable direction.

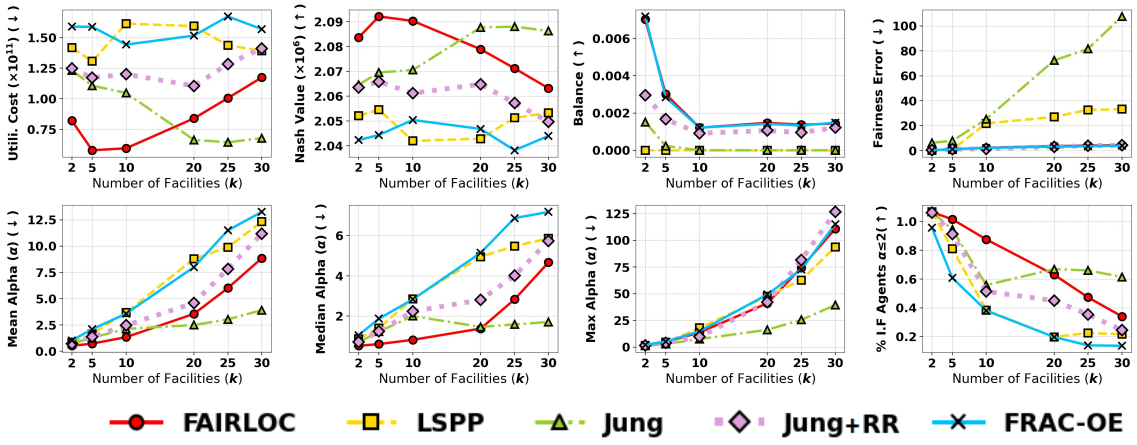


Figure 4.48: The plot shows the variation in metrics across varying k for opening **Schools** in the **US-South** region, with **language** as the protected group. The first row displays methods for utilitarian cost, Nash value, group fairness metrics balance and fairness error. The second row compares the α values distribution using mean, median, max and the number of agents having $\alpha \leq 2$. The arrow on the Y-axis indicates a favorable direction.

Chapter 5

Group Fairness as Capacity Constraints in Online Clustering

Abstract

Clustering is a widely used unsupervised learning tool with applications in numerous real-world problems. Deploying traditional clustering solutions needs to accommodate two major challenges. Firstly, to handle the continuous influx of data points arriving over time. Secondly, these traditional clustering methods can result in highly skewed clusters where one cluster is notably larger than others, rendering them unsuitable for scenarios such as logistics and routing. In response, capacitated clustering approaches have emerged over the past decade. These approaches limit the number of data points each cluster can accommodate, thus resulting in more uniform cluster formations. In an online version of capacitated clustering, the algorithm must make an irrevocable decision for each incoming data point, determining whether to establish it as a new center or allocate it to existing centers. The goal is to minimize the count of opened centers while adhering to capacity constraints and achieving a satisfactory approximation of the clustering cost compared to the optimal solution. Although exploring online capacitated clustering remains uncharted, we are the first to propose a probabilistic algorithm called `COCA` *for h-dimensional euclidean spaces. We theoretically bound the number of centers opened and provide constant cost approximation guarantees. In order to prevent online* `COCA` *from resulting in clusters predominantly with data points belonging to a particular protected group value, we next extend unfair* `COCA` *to obey group fairness constraints. The extended algorithm called* $\texttt{COCA}_F$ *undergoes similar theoretical analysis. The experimental validation on different datasets showcases the efficacy of all proposed algorithms on the number of centers opened and cost.*

5.1 Introduction

Clustering is a widely used tool in data mining and finds practical application in many real-world scenarios, including, but not limited to, automatic resume screening, detecting

A preliminary part of this chapter has appeared in the CODS-COMAD conference 2024 [223] (as Extended Abstract). The work was appreciated by the Best Paper Award (Runner's Up) in the Young Researcher's Symposium at the conference. A detailed version of this chapter is published in the European Conference on Artificial Intelligence (ECAI) 2024 [224].

fraudulent claims, and targeted advertisements [225]. The past decade has witnessed various notable clustering methods such as k-means, k-medoid, k-median, and k-center [226, 227]. The fundamental principle that underlies these methods is partitioning the data points into k distinct groups (called clusters) such that data points within the same cluster are more similar than others. In centroid based clustering each cluster is represented by a center. Further, the similarity of data points to the center is measured with the help of different distance metrics such as Euclidean, Manhattan distance, etc. The objective of cluster formation varies across these methods; for example, k-means seeks to minimize the sum of square distances (p=2 norm; Euclidean distance) between the data points and their respective centers [1]. In contrast, k-median and k-center minimize the sum of absolute distances ($p = 1$ norm) and the maximum distance within a cluster, respectively [228].

However, these traditional methods do not impose restrictions on the sizes of the clusters, leading to clusters with arbitrary sizes. This lack of constraint can result in highly skewed clusters, where one cluster is significantly larger than the others with small sizes, thus hampering their applicability to real-world problems. For example, in logistics distribution (stores/garbage) or workforce team formulation, capacity constraints are defined by the number of customers (or employees) an individual salesperson (or manager) can serve. This poses management and productivity challenges [22, 229, 230]. To address the need for more uniform cluster sizes, researchers have delved into clustering with size constraints i.e., 'capacitated clustering' [231].

The capacitated clustering algorithms can be categorized based on data access and applications, dividing them into offline, streaming, and online environments. In the offline environment, all the data points are known in advance and are available in memory. This model provides the most flexibility in terms of data availability and finds application in fields such as group team formations, student project teams, facility location, and employee allocation [232, 165]. However, the scalability of these offline solutions is constrained by the size of the main memory. In contrast, streaming environments divide data into chunks that can easily fit into the memory. The performance of such algorithms is compared based on the number of passes performed over the complete data points [233]. Existing state-of-the-art (`SOTA`) approaches in capacitated streaming are reviewed in [36, 234].

Online clustering is a more stringent variation of these environments, where an endless stream of data points arrives over time. Due to limited memory, the algorithm must make an irrevocable decision about incorporating an incoming data point into existing clusters or opening it as a new center. Once a data point becomes a center, it remains so forever. Similarly, any data point previously seen cannot be chosen as the center when a new data point arrives [235, 236, 237, 238, 38, 39]. An important aspect to note in online clustering pertains to the absence of information regarding the ordering of arrival of data points in the stream. As a result, the algorithm ends up opening more number of centers ($k_{\texttt{actual}}$) than the desired target ($k_{\texttt{target}}$), i.e., $k_{\texttt{actual}} \geq k_{\texttt{target}}$ to maintain good approximation guarantees on objective cost. In online capacitated clustering, all these constraints are

[1] referred to as objective or clustering cost interchangeably in literature.

imposed while adhering to a given capacity requirements. Note that $k_{\texttt{target}}$ and k are used interchangeably for ease of reading.

To understand the need for online capacitated clustering, consider the dynamic landscape of wholesale distribution networks. In this scenario, retailers employ salespersons who navigate cities to promote products, offer discounts, and build relationships with consumers [22]. To enhance consumer retention, it becomes imperative to provide specialized salespersons. Efficient market coverage is achieved by clustering consumers (shopkeepers and direct customers) based on various features such as product consumption, order volume, and location [22]. The resulting clusters group similar consumers together for personalized marketing. However, a crucial limitation arises in the form of workload constraints, with each salesperson having a maximum capacity to maintain a healthy work-life balance and also offer quality service. Furthermore, the continuous influx of new consumers in a growing market makes handling such a vast network challenging. Traditional offline solutions face computational hurdles in adapting to these changes, emphasizing the need for online solutions. Our example acknowledges that decisions made in online solutions, such as salesperson and consumer allocations, are irrevocable. This permanence is vital as salespersons develop trust and liaisons with consumers over time, and making changes to assignments is impractical and potentially detrimental to established relationships.

This necessitates investigating online clustering solutions that incorporate capacity constraints. Given that this is a comparatively challenging and hard problem [229], only a few works are available in one [239, 240] and two-dimensional [241] space. These works specifically address the k-center objective and exploit the geometrical structural properties of one, two-dimensional spaces to devise deterministic algorithms. However, they are not directly extendible to higher dimensional spaces and alternative objective functions such as k-means or k-median, which focus on minimizing the distance between each data point assignment and its center. To this, we propose a probabilistic approach that handles capacity constraints and works well for any h-dimensional Euclidean spaces similar to Liberty et al. [38], Bhaskara and Ruwanpathirana [39] available in uncapacitated online clustering. This chapter addresses the problem of minimizing the clustering objective while satisfying the capacity constraints in an online setting[2]. The challenge arises when there is an upper limit on the number of cluster centers that can be opened; either many data points are assigned to a single (or a few) cluster(s), resulting in skewed clustering and a violation of capacity constraints, or an inefficient assignment, leading to high objective costs. Our proposed algorithm (`COCA`) addresses this problem by randomized assignments. After a certain initial number of centers are created, with probability $(1 - p_t)$, each incoming t^{th} data point is assigned to the closest available center with remaining capacity (see Algorithm 8) and with probability p_t is designated as a new center.

Since we impose capacity constraints on the overall size of the clusters, analyzing the

[2]With unrestricted capacity constraints, our problem reduces to the problem of uncapacitated online clustering.

distribution of data points from different protected group values (say, male and female in gender) in each cluster is essential. The existing clusters may exhibit an imbalance, and deploying such biased clusters can have substantial societal implications. For example, in our running example, each salesperson offers discounts to their clusters. We must ensure that customers from diverse group values have equitable and fair access to all available offers. One way to solve this problem is by imposing group fairness constraints on each cluster. To this, we propose an extension of COCA called $\texttt{COCA}_F$ that handles the continuous flow of data points while obeying group fairness constraints. As in the online setup, no prior information is available about ordering the data points, thus ensuring minimum thresholds on data points from different group values such as minority protection [12] and $\boldsymbol{\tau}$-ratio fairness [44] will not work. Instead, motivated by restricted dominance [12, 92], we control the over-representation of data points in any cluster and impose capacity constraints on data points from different group values in each cluster. Similar to the unfair version, $\texttt{COCA}_F$ makes probabilistic decisions for each incoming data point of whether the new data point should be opened as a center or assigned to existing centers. Thus, to summarize, the following are our contributions to both unfair and fair online clustering:

Contributions in Unfair Online Capacitated Clustering:

1. With careful choice of p_t, we establish an upper bound on the number of centers opened by COCA, that matches with that of the uncapacitated setting.

2. We provide a constant approximation guarantee for the objective cost compared to optimal offline capacitated clustering. These guarantee enhances existing bounds in an uncapacitated setting [38] by a logarithmic factor.

3. We estimate the challenging, a-priori unknown total number of data points using the doubling trick.

4. We establish an interesting connection between our framework and a well-known coupon collector problem to determine the initial number of centers to be opened.

5. Empirical evaluations demonstrate the comparable performance of COCA with existing SOTA in uncapacitated online clustering on variety of datasets.

Contributions in Fair Online Capacitated Clustering:

1. We are the first to model group fairness in online clustering as capacity constraints.

2. We establish an upper bound on the number of centers opened by $\texttt{COCA}_F$ and provide cost approximation for cost compared to optimal fair offline clustering.

3. $\texttt{COCA}_F$ undergoes experimental validation on synthetic and real-world datasets. Results showcase a small trade-off in online cost compared to offline group fair clustering while obeying capacity and group fairness constraints.

Organization: Section 5.2 reviews the literature. Section 5.3 outlines the preliminaries needed for the paper. Section 5.4 and 5.5 present the proposed unfair online algorithms. Next, we extend the proposed algorithms to obey group fairness constraints in Section 5.6. Section 5.7 then evaluates the efficacy of proposed algorithms experimentally. Finally, Section 5.8 concludes with potential future directions.

5.2 Related Work

The existing clustering literature encompasses various methods, ranging from hierarchical to centroid-based. This work focuses on centroid-based clustering due to its computational efficiency, scalability[3], and interpretability[4]. We now review various `SOTA` approaches to approximate the capacitated clustering problem available in different environments based on data access and applicability.

Offline Capacitated Clustering: The first attempt in offline capacitated clustering problem is by Mulvey and Beck [242]. The authors proposed a heuristic that modifies uncapacitated clustering by validating capacity constraints before assignment. Building on this work, Mai et al. [243] extended the heuristic method. Later, Boccia et al. [244] proposed a more effective and exact solution using cutting plane algorithms. The complete list of offline capacitated clustering approaches is available in [245, 246, 247]. Notably, the best approximation factor for the capacitated k-means/k-median objective in Euclidean spaces is $(1+\epsilon)$ [232] where $\epsilon > 0$, and for the k-center method, it is two [248].

Streaming Capacitated Clustering: An initial attempt to achieve uniform (almost equal-sized) clustering is by Bateni et al. [249]. The algorithm they propose requires three passes over the data stream. Later, Esfandiari et al. [250] improves the work and proposes a single-pass algorithm. However, their algorithm is not directly applicable to the online environment as it involves generating coresets first and then obtaining the final assignment. In contrast, in the current online setting, decisions must be made as soon as the data point arrives.

Online Capacitated Clustering: Recent investigations into k-center problem have explored the one-dimensional case [239, 240]. In these works, each cluster is a closed interval with no restriction on the cluster's diameter. Whenever a data point falls in a specific interval, that interval opens as a cluster for future data points. The goal is to minimize the sum of the diameter of clusters while accommodating all data points. Extending this concept to two-dimensional space involves replacing intervals with squares [241]. The algorithm initiates a new cluster whenever a data point falls within an unopened square-grid cell, and the goal is to reduce the sum of the area of the opened clusters. However, the study of k-means or k-median objective, especially in higher dimensions, remains an open problem, a concern we tackle in this chapter.

Deep and Contrastive clustering: Deep clustering methods require model training with data before responding to online queries [251, 252, 253]. In contrast, our setting is

[3]In terms of dataset size and dimensionality.
[4]In terms of visualization and interpretation.

much stricter, and mini-batches of samples for training may not be available. Works that employ contrastive clustering also face similar limitations [254, 255].

Fairness in Clustering: Recent studies have revealed that the clusters stemming from the above algorithms may not exhibit a sufficient representation of different protected groups (say gender) within each cluster. An attempt to tackle such demographic bias in offline capacitated clustering is by Le Quy et al. [165], Tran et al. [256]. The authors impose additional constraints using the concept of `Balance` which requires each protected group value to have approximately equal representation in every cluster [42]. Similarly, works in student topic grouping problems devise knapsack-based reduction or fair coresets to achieve the maximum possible `Balance`. Although a prior work reformulates the `Balance` concept by utilizing linear programming to set upper bounds on the number of data points from each group [92], this extension does not directly apply to online settings and is limited to k-center. A few studies also examine online facility location [257, 258]. However, these works differ slightly from ours as they either focus on assignment problems without addressing facility location or leverage the benefits of multiple expert advice, which differs from online capacitated clustering.

5.3 Preliminaries

Let $X \subseteq \mathbb{R}^h$ be an endless stream of data points with x_t being the point arriving at time t. Each data point $x_t \in X$ is articulated using h dimensional real-valued features. We assume that these points are embedded in metric space with $d : X \times X \to \mathbb{R}^+ \cup \{0\}$ measuring the dissimilarity between any two data points. Then, the goal of any centroid-based clustering algorithm is to partition data points into clustering $\mathcal{C} = (C, \phi)$. The clustering produces k disjoint subsets ($[k] = \{1, \ldots, k\}$) with centers $C = \{c_j\}_{j=1}^{k}$ using an assignment function $\phi : X \to C$ that maps each data point to corresponding cluster center.

Also, let L_p^*, ϕ^* represent the optimal (offline/online) objective cost (Definition 2.1) and the optimal assignment function, respectively. We now define the capacity constraint mapping $\boldsymbol{\gamma} : C \to [0, 1]$, representing the total capacity on the fraction of data points each cluster accommodates. We consider the following assumptions on capacity constraints:

- Capacities are same across all clusters i.e., $\boldsymbol{\gamma}(c_j) = \gamma,\ \forall j \in [k]$. This ensures equal treatment among clusters, avoiding any favouritism. Furthermore, in the online setting, imposing capacity constraints at the cluster level is infeasible due to the dynamic nature of the number of opened centers.

- With n as the total data points that the algorithm eventually sees, we adopt an assumption that γ is a multiple of n/k, i.e., $\gamma = \frac{\vartheta n}{k}$. Here ϑ indicates the permissible degree of skewness among clusters and belongs to $[1, k]$. When $\vartheta = 1$, it results in perfect uniform cluster sizes, while $\vartheta = k$ indicates an uncapacitated clustering problem.

The first half of the chapter will focus on unfair online settings, and we begin with the initial work, as described by Liberty et al. [38]. The fully online algorithm initiates by

selecting the first $(k+1)$ data points as the initial set of centers to estimate the lower bound on objective cost (ℓ_p^*). The heuristic is based on the idea that clustering $(k+1)$ data points should put at least two data points together. Subsequently, for the remaining data points, the algorithm determines whether to assign each data point to the nearest center ($c \in C$) or if the data point's distance incurs a high assignment cost $d(x_t, c)$. This assessment is quantified using a probability that depends on the ratio of the assignment cost to the center opening cost (f_r). To prevent excessive points from being opened as centers, value of f_r for round r doubles when the center count exceeds a predefined threshold. While Bhaskara and Ruwanpathirana [39] improves this method, the algorithm now makes delayed decisions. This implies that if the current data point needs to be opened as a center, it is not opened immediately but deferred to a later time. Although this delayed approach contributes to improved objective cost approximation by a logarithmic factor, the current focus of the study is on immediate assignment or opening of data points, as necessitated by the need in the running example in the previous Section 5.1, i.e., each new consumer must be promptly assigned to a salesperson to ensure the seamless operation of the business and timely product deliveries. A delayed response from the wholesaler could result in a shift to alternate avenues. Consequently, we build upon the algorithm presented in Liberty et al. [38] by extending it to capacitated clustering. Our approach introduces several modifications that result in substantial enhancements over the conventional online clustering problem (and subsequently to the online capacitated clustering problem).

- Through experimental observation, we have noted that an initial selection of $(k+1)$ data points for estimation of lower bound on optimal cost can potentially result in a higher likelihood of opening more centers in future. It is primarily due to bad cost estimation that the algorithm relies on. Instead, we propose a selection criterion based on the non-uniform coupon collector problem.
- In Liberty et al. [38], algorithm estimates the total data points using the current point count observed, achieving $O(\log n)$ cost approximation. Our approach improves this by updating the estimation with the doubling trick, providing improved constant cost approximation.

We now formally restate **non-uniform coupon collector problem** with replacement Doumas and Papanicolaou [259]:

Claim 5.1. *Given m distinct coupon types, the expected number of coupons required to obtain at least one coupon from each type is denoted as $\mathcal{H}_m$, and it is calculated as follows: $\mathcal{H}_m = \sum_{a=1}^{m} (-1)^{a-1} \sum_{1 \le j_1, \ldots, j_a \le k} \frac{1}{p(j_1)+\ldots+p(j_a)}$ where $p(i)$ is the probability of obtaining a coupon of type i.*

We aim to determine the expected number of data points be opened as a center to ensure representation from each center in the optimal capacitated clustering. As this will result in a better approximation of ℓ_p^*, which the fully online algorithm will require. To achieve this, we employ a non-uniform coupon collector problem as follows: consider coupons to be the

data points and each coupon type to be the centers in the offline optimal clustering (i.e., $m = k$ in our case). However, the main challenge lies in computing the probabilities $p(i)$, representing the probability of a data point belonging to cluster i. When the capacities are uniform, i.e. n/k with $\vartheta = 1$ and data points are coming uniform at random from any cluster, it becomes evident that we obtain $p(i) = 1/k \; \forall i \in [k]$ thus leading to $H_k = k \log k$. Further, since $p(i)'s$ are not known to us, we restrict the value of H_k to be $k \log k$, and therefore, the total number of centers opened by COCA remains of the same order as that of by Liberty et al. [38]. It must be noted that, in order to satisfy Claim 5.1, however, H_k may reach the value of n (when differences in $p(i)$'s are arbitrarily high), which is again consistent with the literature as shown by Moshkovitz [260] that even with knowledge of n, any algorithm would inevitably open $\Omega(n)$ centers in the worst case ordering of data points. Note that our theoretical proofs hold and remain unaffected by choice of $p(i)$'s and the value of H_k. It's just that if prior information about sampling probabilities is known, one can leverage Claim 5.1 to obtain a better estimate of initial centers (and ℓ_p^*).

5.4 Capacitated Semi-Online Clustering Algorithm (CSCA)

We begin by looking into semi-online clustering called Capacitated Semi-online Clustering Algorithm (CSCA) wherein the total number of data points (n) and lower-bound on optimal cost (ℓ_p^*) is known. Note that most restrictions in fully online clustering (i.e., when both these n, ℓ_p^* are unknown) apply to semi-online clustering. This means that for each data point, the algorithm must make an irrevocable decision to either assign it to the existing centers or open it as a new center. The complete pseudo-code for CSCA is described in Algorithm 7. CSCA method begins by initializing the capacity vector (Γ) and assignment function (ϕ) as an empty set. It also sets the round counter (r) to one and the estimate of the number of centers opened in the current round (q_r) to 0 in lines 1 to 2. Since ℓ_p^* is given CSCA can compute the center opening cost (f_r) by plugging the value of the lower bound (ℓ_p^*), k, and n to get $f_r = \ell_p^* \vartheta / k \log n$ in line number 4. This helps avoid opening up a few initial centers to estimate ℓ_p^*, and CSCA can proceed to execution just by opening the first data point as center (line 3). Now, for all the remaining incoming data points, the method takes a probabilistic decision about opening a data point as a center from lines 5 to 20. The decision considers the ratio between the distance to the closest center with remaining capacity ($d(x_t, c)$) and the center opening cost (f_r). This is because if the distance to the closest available center will dominate over the center opening cost, the probability value $p_t = d(x_t, c)/f_r$ will increase and vice versa. In case there is no closest available empty center then the incoming data point will be opened as center. However, to control too many centers to open up CSCA doubles the center opening cost if the current estimate of centers opened (q_r) in any round r rises above a certain threshold and resets the estimate counters for the next round ($r + 1$) in lines 15 to 18. However, if the event of opening a data point is unsuccessful (with probability $(1 - p_t)$), then CSCA assigns the data point to the closest center. We now look into CSCA's theoretical guarantees.

Algorithm 7: Capacitated Semi-online Clustering Algorithm

Input: set of n data points X, optimal ℓ_p^*, and capacity constraint γ
Output: cluster centers C and assignment function ϕ

Initialize $\Gamma \leftarrow \varnothing$
Initialize $\phi \leftarrow \varnothing$, $r \leftarrow 1$, $q_r \leftarrow 0$
Open first data point as center (c_1) and set $\Gamma(c_1) = \gamma$, $q_r \leftarrow 1$
Initialize center opening cost $f_r = \ell_p^* \vartheta / k \log n$
for *each remaining* $x_t \in X$ **do**
    $c \leftarrow \text{argmin}_{c \in C : \Gamma(c) > 0} \, d(x_t, c)$ // Find the closest center
    With probability $p_t = \min\left(d(x_t, c)/f_r, 1\right)$:
        $C \leftarrow C \cup \{x_t\}$ // Open the data point as center
        $\phi(x_t) = x_t$ //Set the assignment function
        $\Gamma(x_t) = \gamma$ //Initializing the capacity constraints
        $q_r \leftarrow q_r + 1$
    Otherwise, with probability $1 - p_t$:
        $\phi(x_t) = c$ //Assign the data point to center
        $\Gamma(c) = \Gamma(c) - 1$ // Update capacity
    if $q_r \geq \frac{3k}{\vartheta}(1 + \log n)$ **then**
        $r \leftarrow r + 1$
        $q_r \leftarrow 0$
        $f_r \leftarrow 2 f_{r-1}$ // Update the opening cost
    end
end
return (C, ϕ)

5.4.1 Theoretical Results

We now first look into the expected number of centers opened by Algorithm 7, and subsequently, cost approximation bounds. To this, let us denote optimal clustering as $\mathcal{C}$ with corresponding clusters $\{\mathcal{C}_1^*, \ldots, \mathcal{C}_k^*\}$ and assignment function ϕ^*. We omit the p-norm factor from the distance function in the proofs for ease of reading. However, proofs hold for all finite values of p.

Theorem 5.2. *Let C be a set of cluster centers opened by Algorithm 7. Then,* $\mathbb{E}(|C|) = O\left(\frac{k}{\vartheta} \log(n) \log\left(\frac{L_p^*}{\ell_p^*}\right) + k \log(n)\right)$.

Proof. Let $L_{p,i}^*$ be the optimal capacitated clustering cost of cluster i and is given by $L_{p,i}^* = \sum_{x \in \mathcal{C}_i^*} d(x, \phi^*(x))$. So the total optimal cost is $L_p^* = \sum_{i=1}^k L_{p,i}^*$. Further, let A_i^* denote the average distance from data points in the i^{th} optimal cluster to its center and is computed as $A_i^* = \frac{1}{|\mathcal{C}_i^*|} \sum_{x \in \mathcal{C}_i^*} d(x, \phi^*(x)) = L_{p,i}^* / |\mathcal{C}_i^*|$.

Now, our primary goal is to bound the number of centers opened. We have k optimal clusters, and as the arrival of data points is unknown in the online setup, we end up opening more centers in each cluster as an estimation of the optimal center. Let us now divide the k optimal clusters into different rings motivated from [261, 262, 263]. The broader idea is to compute the expected number of centers that we end up opening in each of these rings. The 0^{th} ring is denoted by $\mathcal{C}_{i,0}^* = \{x \in \mathcal{C}_i^* : d(x, \phi^*(x)) \leq A_i^*\}$. The

subsequent rings, from 1 to τ, are given by $\mathcal{C}^*_{i,\tau}$=$\{x \in \mathcal{C}^*_i : 2^{\tau-1}A^*_i < d(x, \phi^*(x)) \leq 2^\tau A^*_i\}$. Note that a cluster $\mathcal{C}^*_i$ will be divided into $(1 + \log n)$ rings, as all rings after $\log n$ will be essentially empty. Let r' be the first round when the center opening cost $f_{r'}$ becomes some fraction of L^*_p such that, $f_{r'} \geq \frac{2^4 L^*_p \vartheta}{k \log n}$. Now, we bound the expected number of centers in two separate parts, i.e., before round r' and second during and after round r'. Let us first begin with the former,

Case 1: By the definition of r', we have $f_{r'-1} < \frac{2^4 L^*_p \vartheta}{k \log n}$. Further, since the center opening cost becomes twice at every round, we have, $f_{r'-1} = 2^{r'-1} f_1$. Substituting the value of $f_1 = \frac{\ell^*_p \vartheta}{k \log(n)}$, we get, $r' \leq \log\left(\frac{L^*_p}{\ell^*_p}\right) + 5$. Therefore, before round r', the number of centers opened by the algorithm is,

$$\mathbb{E}(|C|_{\text{before } r'}) = O\left(\frac{3k}{\vartheta}(1 + \log n) \log\left(\frac{L^*_p}{\ell^*_p}\right)\right) \tag{5.1}$$

Case 2: Now, let's look into computing the number of centers opened during and after round r' in each of these rings. To avoid getting struck due to not knowing order of arrival of data points, we will loosely estimate the expected number of centers present in any ring during or after round r'. To this, we divide the bounds into three subparts-

Case 2(a): First, we estimate the number of new centers that will open for the first time in each ring. Let's denote these centers as K^1_τ. Since there are a total of $(1 + \log n)$ rings in each cluster, therefore the total number of such centers are $\sum_k \sum_\tau 1 = k(1 + \log n)$.

Case 2(b): Next, suppose there is a data point x that arrives and the closest center to x has already reached its capacity; in such a case, the data point will continue searching for the next closest center in any of the rings in increasing order of distance. There are two possibilities: either data point x will find a vacant center or its likelihood of becoming a center increases as it delves further into the chain if the next closest center is too far away (handled in next case). Let's denote the extra number of centers that need to be opened up in any ring $\boldsymbol{\tau}$ due to the exhausting capacity of the first centers of Case 2a, be denoted by K^c_τ. It is important to note that once a center K^1_τ fills up and we need to open a second center within the ring, then at least $\frac{\vartheta n}{k}$ data points have already arrived and been assigned. Therefore, the total number of such K^c_τ centers over all rings and clusters is upper bounded by k/ϑ i.e., $\sum K^c_\tau \leq k/\vartheta$.

Case 2(c): Now since there were two possibilities: either data point x will find a vacant center or its likelihood of becoming a center increases as it delves further into the chain if the next closest center is too far away. Now, the remaining task is to bound the probabilistically opened centers in each ring apart from the centers opened in the previous two subcases. To do this according to Algorithm 7, if a data point x is the initial center opened within any ring, then the probability of subsequent point x' from the same ring opening as a center is defined and bounded using the properties of rings as follows:

$$\frac{d(x, x')}{f_{r'}} \leq \frac{d(x, \phi^*(x)) + d(x', \phi^*(x))}{f_{r'}} \leq 2 \cdot 2^\tau \frac{A^*_i}{f_{r'}}$$

($\because$ using triangular inequality and ring property)

So, the expected number of centers that will open in any ring over all rounds $r \geq r'$ is $\sum_{r \geq r'} \frac{2 \cdot 2^{\tau} A_i^*}{f_r} |\mathcal{C}_{i,\tau,r}^*|$. Summing these probabilistic centers over all rings and using $f_r \geq f_{r'}$ we obtain the number of centers opened for estimating one optimal cluster center as,

$$\begin{aligned} K_{\boldsymbol{\tau}}^p &= \sum_{\tau \geq 0} \left(\sum_{r \geq r'} \frac{2 \cdot 2^{\tau} A_i^*}{f_r} |\mathcal{C}_{i,\tau,r}^*| \right) \leq \sum_{\tau \geq 0} \frac{2 \cdot 2^{\tau} A_i^*}{f_{r'}} \sum_{r \geq r'} |\mathcal{C}_{i,\tau,r}^*| \\ &\leq \sum_{\tau \geq 0} \frac{2 \cdot 2^{\tau} A_i^*}{f_{r'}} |\mathcal{C}_{i,\tau}^*| \leq \frac{2 A_i^* |\mathcal{C}_{i,0}^*|}{f_{r'}} + \frac{4}{f_{r'}} \sum_{\tau \geq 1} \sum_{x \in \mathcal{C}_{i,\tau}^*} 2^{\tau - 1} A_i^* \\ &\leq \frac{6 L_{p,i}^*}{f_{r'}} \qquad \text{(Using } L_{p,i}^* = A_i^* |\mathcal{C}_i^*| \text{ and } 2^{\tau-1} A_i^* \leq d(x, \phi^*(x))\text{)} \end{aligned}$$

Summing this up for all k cluster centers and considering the estimate of $f_{r'} \geq \frac{16 L_p^* \vartheta}{k \log n}$ we get,

$$K_k^p \leq \frac{6 L_p^*}{f_{r'}} \leq \frac{6 k \log n}{16 \vartheta} \tag{5.2}$$

Therefore, number of total centers opened in Case 2 are as follows

$$\mathbb{E}(|C|_{\text{during and after } r'}) = O\left(K_{\boldsymbol{\tau}}^1 + \sum_{\boldsymbol{\tau}, k} K_{\boldsymbol{\tau}}^c + K_k^p \right)$$

$$= O\left(k(1 + \log n) + \frac{k}{\vartheta} + \frac{k \ log(n)}{\vartheta} \right) = O\left(\frac{k}{\vartheta} log(n) \right) \tag{5.3}$$

Thus, combining Equation 5.1 and 5.3, completes the proof, resulting in the total expected number of centers opened by `CSCA` as $O\left(\frac{k}{\vartheta} \log(n) \log\left(\frac{L_p^*}{\ell_p^*} \right) \right)$. Note that for the unbounded capacity case, when $\vartheta = k$, our bounds in semi-online algorithm `CSCA` match with that of Liberty et al [38].

□

Theorem 5.3. *Let $L_p^{\texttt{CSCA}}$ represent the cost of the semi-online capacitated cost and L_p^* denote the optimal offline capacitated cost. Then, $\mathbb{E}\left(L_p^{\texttt{CSCA}}\right) = O\left(L_p^*\right)$.*

Proof. To approximate the cost guarantees, our primary focus is on bounding the assignments in line 13 in `CSCA`. In all other assignments, data points are centers themselves, resulting in zero cost. However, after the opening of these initial centers, the data points have two possibilities of getting assigned. Firstly, they may be assigned to one of the centers within the same ring as the data point's optimal ring. Secondly, suppose the center within the same ring is already occupied; in that case, data points may be assigned to a center located in a different ring within the same cluster or in a ring belonging to a different cluster. We first bound the latter as follows:

Case 1: Note that the cost of data points (say $x_t \in X$) going to rings other than the optimal one incurs a cost equal to the distance to the assigned center from set C. We will

use the Lemma 1 of Liberty et al. [38] to bound these costs, i.e., $\mathbb{E}(d(x_t, C))$ and restate the lemma below:

Lemma 5.4 (Liberty et al. [38])**.** *Given a sequence of n independent experiments, each of which succeeds with probability atleast $\min(A_i/B, 1)$ where $B \geq 0$ and $A_i \geq 0\ \forall i \in [n]$. Let t be the (random) number of sequential unsuccessful experiments, then, $\mathbb{E}(\sum_{i=0}^{t} A_i) \leq B$.*

Now, before we delve into using the above lemma, let us first understand the mapping between our problem and the technical lemma. In `CSCA` probability of event (center opening) is at least $\min(d(x_t, C)/f_{r'}, 1)$ where x_t is any data point and C is set of existing vacant centers. On using the fact that given R as the last round, $f_R > f_{r'}$ for any round $r' < R$, the denominator can be made constant as in the lemma. Now, each independent unsuccessful experiment represents the assignment of one data point, and we can use the lemma to bound the expected value of the sum of A_i's ($d(x_i, C)$'s in our problem) by $B(= f_R)$. Note that once the event gets successful, i.e., the center in any ring gets opened, we will bound the cost of assignments to the opened center in the next case, but here we look into the scenario once this center gets filled up. In such a situation, assignments are again upper bounded by $O(f_R)$ along similar lines.

Case 2: We will next bound the cost of all data points that are allocated within the ring. Now, after any data point x is opened as center, then cost of subsequent point x' is given by $d(x, x') \leq d(x, \phi^*(x)) + d(x', \phi^*(x)) \leq 2(2^\tau A_i^*)$ (using triangular inequality and ring property). Now, using Equation 5.2, the total cost over all such x' is given as $\sum_{x'} d(x, x') \leq 6L_p^*$. It is important to note that, unlike the uncapacitated case in Liberty et al. [38], once a center gets opened in the ring, its capacity can eventually get exhausted, and then one returns to Case 1 and needs to wait until the next center is opened within the ring and once a new center opens up, which is already accounted for by Case 2. Therefore, the total expected cost by combining both cases over all rings is given as follows:

$$O(f_R k \log n + L_p^*) \tag{5.4}$$

Therefore, we must find our case's expected value of f_R. To this, let us consider some round r' in `CSCA` such that,

$$f_{r'} \geq \frac{16 L_p^* \vartheta}{k \log n} \geq \frac{16 L_p^* \vartheta}{k(1 + \log n)} \tag{5.5}$$

Using Equation 5.2 (i.e., $6L_p^*/f_r'$), Equation 5.5 and Markov inequality, the probability of opening more than $\frac{3k}{\vartheta}(1 + \log n)$ centers is $\frac{1}{8}$ and thus, `CSCA` concluding at round r' is equal to $\frac{7}{8}$. Now, let b be probability that `CSCA` terminates before round r', then,

$$\begin{aligned}
\mathbb{E}(f_R) &\leq b f_{r'-1} + (1-b) \sum_{r=r'}^{\infty} f_r \left(\frac{7}{8}\right) \left(\frac{1}{8}\right)^{(r-r')} \\
&\leq b f_{r'} + \frac{7}{8}(1-b) \sum_{i=0}^{\infty} f_{r'+i} \left(\frac{1}{8}\right)^{i} < O(f_{r'}) \qquad \text{(using } f_{r'+i} = 2^i f_{r'} \text{ and } \tfrac{1}{8} < 1)
\end{aligned}$$

$\implies \mathbb{E}(L_p^{\texttt{CSCA}}) = O(f_R k \log n + L_p^*) = O(L_p^*)$.

□

5.5 Capacitated Online Clustering Algorithm (COCA)

Now, we delve into a fully online setup in which n is unknown, and the algorithm needs to compute the lower-bound ℓ_p^* without any prior knowledge of n. To approximate ℓ_p^*, the algorithm leverages the insight that once H_k ($\geq k$) data points are opened as center, a more accurate estimation of ℓ_p^* can be obtained by performing clustering on these H_k data points (see Claim 5.1). Further, for monitoring the estimated value of n, the method utilizes a doubling technique in lines 22 to 23, wherein the estimate is doubled once it is achieved. The code is outlined in Algorithm 8. Also, for better understanding, the complete algorithm is pictorially illustrated as an image flow diagram in Figure 5.1.

Algorithm 8: Fully online COCA

Input: set of datapoints X and capacity constraint γ
Output: cluster centers C and assignment function ϕ

1 Initialize $\Gamma \leftarrow \varnothing$ `// stores vacant capacity of center`
2 Open first H_k points as centers (Claim 5.1) and $\forall j \in [H_k]$ set $\Gamma(c_j) = \gamma$
3 Initialize $\phi \leftarrow \varnothing$, $r \leftarrow 1$, $n_r \leftarrow H_k$, $q_r \leftarrow H_k$, idx $\leftarrow H_k$
4 $\ell_p^* \leftarrow$ objective cost on H_k using Mulvey and Beck [242].
5 Initialize center opening cost $f_r = (\ell_p^* \vartheta)/(k \log n_r)$
6 **for** *each remaining* $x_t \in X$ **do**
7 $c \leftarrow \text{argmin}_{c \in C : \Gamma(c) > 0} d(x_t, c)$
8 **With probability** $p_t = \min\left(d(x_t, c)/f_r, 1\right)$:
9 $C \leftarrow C \cup \{x_t\}$
10 $\phi(x_t) = x_t$
11 $\Gamma(x_t) = \gamma$
12 $q_r \leftarrow q_r + 1$
13 **Otherwise, with probability** $1 - p_t$:
14 $\phi(x_t) = c$
15 $\Gamma(c) = \Gamma(c) - 1$
16 **if** $q_r \geq \frac{3k}{\vartheta}(1 + \log n_r)$ **then**
17 $r \leftarrow r + 1$
18 $q_r \leftarrow 0$
19 $f_r \leftarrow 2 f_{r-1}$
20 **end**
21 idx $\leftarrow$ idx $+ 1$
22 **if** $idx \geq n_r$ **then**
23 $n_r \leftarrow 2 n_r$
24 **end**
25 **end**
26 **return** (C, ϕ)

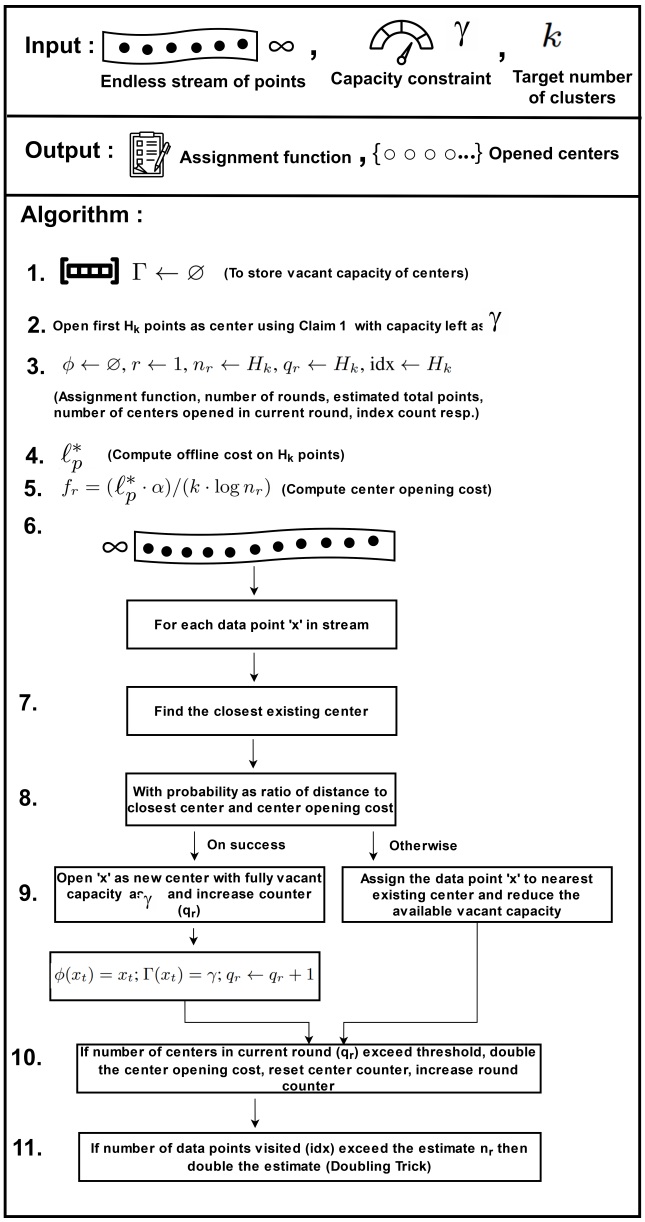


Figure 5.1: Image flow for proposed online method COCA.

5.5.1 Theoretical Results

Theorem 5.5. *If C is set of centers opened by COCA, then, $\mathbb{E}(|C|) = O\left(H_k + \frac{k}{\vartheta}\log(n)\log(n\delta)\right)$ where $\delta = \frac{\max_{x,x'} d(x,x')}{\min_{x,x':x\neq x'} d(x,x')}$.*

Proof. The proof will follow similarly to Theorem 5.2 except for the fact that in each round instead of opening $\frac{3k}{\vartheta}(1 + log\ n)$ centers, we are opening $\frac{3k}{\vartheta}(1 + log\ n_r) \leq \frac{3k}{\vartheta}(1 + log\ n)$ centers for all r except the last round. Even for the last round $n_r \leq 2n$. Therefore, we can simply substitute the value of ℓ_p^* and L_p^*. Now, as Algorithm 8 computes ℓ_p^* as capacitated cost using Mulvey and Beck [242] on H_k points. So, $\ell_p^* \geq \min_{x,x'\in X:x\neq x'} d(x,x')$. Similarly, the optimal capacitated cost $L_p^* \leq n \max_{x,x'\in X} d(x,x')$ (as the maximum distance from any center (data point) to other points is bounded by the maximum pairwise distance). Substituting these values and adding initial H_k centers completes the proof. □

Theorem 5.5 indicates that the number of centers can be negatively affected by the presence of the term H_k. However, our experiments demonstrate that selecting the initial H_k data points as center, rather than $(k+1)$ (as done in Liberty et al. [38]), actually contributes to opening overall fewer centers because it results in a better estimate of ℓ_p^*. Further, when $k \geq 2$ and n is unknown, Moshkovitz [260] shows that at least $\Theta(\log n)$ centers for random ordering are needed to achieve constant cost approximation. Our upper bound in the online capacitated setting ($\vartheta = k$) aligns with lower bounds in the uncapacitated setting.

Theorem 5.6. *Let $L_p^{\texttt{COCA}}$ be the cost of online Algorithm 8 and L_p^* be the optimal offline capacitated cost. Then, $\mathbb{E}(L_p^{\texttt{COCA}}) = O\left(L_p^*\right)$.*

Proof. We begin with Equation 5.4 given in Theorem 5.3 i.e, $\mathbb{E}(L_p^{\texttt{COCA}}) = O(f_R k \log n + L_p^*)$. Thus, we need to estimate the value of f_R at the last round R. Let us consider any round r such that $f_r \geq \frac{16 L_p^* \vartheta}{k \log(n_r)}$. Then, number of centers opened in round r is given as $q_r \leq \frac{k}{\vartheta}(1 + \log(n_r)) + q_r'$. Here, we pessimistically count one (first) centers in each ring up to round r and q_r' is the number of centers opened in rings after opening former $\lceil k/\vartheta \rceil$ centers. In order to have more rounds than r, COCA needs $q_r' \geq \frac{2k}{\vartheta}(1+\log(n_r))$. We will now compute the probability that COCA terminates by round r. Applying Markov inequality by using the above information along with $\mathbb{E}(q_r') \leq 6L_p^*/f_r$ from Equation 5.2, we get the probability of reaching the next round as at most $3/16$. Thus, if b is the probability that COCA terminates before round r. We have,

$$\begin{aligned}
\mathbb{E}(f_R) &= b f_{r-1} + (1-b) \sum_{r'=r}^{\infty} f_r \left(\frac{13}{16}\right) \left(\frac{3}{16}\right)^{r'-r} \\
&< b f_r + f_r (1-b) \left(\frac{13}{16}\right) \sum_{i=0}^{\infty} 2^i \left(\frac{3}{16}\right)^i \\
&= O(f_r) = O\left(\frac{16 \vartheta L_p^*}{k \log(n_r)}\right). \qquad \text{(using } f_{r-1} = 2f_r \text{ and } b \leq 1\text{)}
\end{aligned}$$

On substituting this back, we get

$$\mathbb{E}(L_p^{\texttt{COCA}}) = O(f_R k \log(n) + L_p^*) = O\left(\frac{16 L_p^* \vartheta k \log n}{k \log n_r} + L_p^*\right)$$

Now since with high probability the algorithm will terminate at r^{th} round and from doubling trick, we can say, $n_r \geq n$. So, $\mathbb{E}(L_p^{\texttt{COCA}}) = O\left(L_p^* \vartheta + L_p^*\right) = O(L_p^*)$. This completes the proof. The derived bounds exhibit a substantial reduction by a logarithmic factor compared to Liberty et al. [38]. Note that, due to capacity constraints in an online setup, there may be some miss-assignments compared to the offline method. However, these disruptions will be minimal owing to constant cost bounds. Additionally, all results hold for any scalar distance metric and for higher dimensions: manhattan or fractional norms [264] are sometimes preferred over Euclidean.

□

Next, we extend our proposed algorithms to accommodate group fairness constraints. Since in online clustering, the total number of data points is not known in advance, so ensuring minimum representation seems less practical from an implementation perspective. Therefore, we model group fairness by limiting the over-representation of any group value in a cluster. This automatically provides a fair chance for data points from other group values. In order to control over-representation in each cluster, we model the problem as online clustering with capacity constraints on each protected group value in every cluster. In the next section, we will provide more details about the proposed fair algorithm.

5.6 Fair Capacitated Online Clustering Algorithm (COCA$_F$)

In this section, we now extend the COCA to a fair version called COCA$_F$. The main changes from COCA involve having separate capacity constraints (γ_a $\forall a \in [m]$). Owing to similar reasons as COCA we keep γ_a same across all the opened centers. Now, since the data points belonging to different group values can arrive in any order, we have separate center opening costs for each group value. This will prevent assigning data points arriving late in the stream to be also open as centers if they are not close enough to existing centers (based on the p_t value). Further, we also now estimate separately the number of data points from each group value using the doubling trick. The complete pseudo-code for the method is provided in Algorithm 9. We now discuss the theoretical results for COCA$_F$.

5.6.1 Theoretical Results

We now first look into the expected number of centers opened by Algorithm 9, and subsequently, cost approximation bounds. Primarily, the main changes in proof arise from the fact that in COCA$_F$, we now have center opening cost for each round r different for each group value, i.e., $f_{a,r}$ where $a \in [m]$ instead of a common f_r. Further, the proofs will now involve bounding the number of centers and cost separately for each group value and then computing the complete results. We first summarize the result on the number of centers opened by COCA$_F$ using the following theorem:

Theorem 5.7. *If C is set of centers opened by COCA$_F$, then, $\mathbb{E}(|C|) = O\left(H_k + \frac{3mk}{\vartheta_a}(1+\log n_a)\log(n\delta) + \frac{mk}{\min_{a\in[m]}(\vartheta_a)} + \frac{6mk\log n_b}{16\vartheta_b}\right)$ where $\delta = \frac{\max_{x,x'} d(x,x')}{\min_{x,x':x\neq x'} d(x,x')}$ and $a \in [m]$ is the most dominant group value in dataset and $b \in [m]$ is the least dominant group value.*

Proof. On similar lines as the previous proofs, suppose that $\mathcal{C}^*_{i,a}$ denotes the clustering of data points in clustering $\mathcal{C}^*_i$ belonging to group value $a \in [m]$. Further let $L^*_{p,i,a}$ be the optimal capacitated clustering cost of cluster i and group value $a \in [m]$ and is given by $L^*_{p,i,a} = \sum_{x_a \in \mathcal{C}^*_{i,a}} d(x_a, \phi^*(x_a))$. So the total optimal cost is $L^*_p = \sum_k \sum_a L^*_{p,i,a} = \sum_{i=1}^k L^*_{p,i}$. Further, let $A^*_{i,a}$ denote the average distance from data points belonging to group value $a \in [m]$ in the i^{th} optimal cluster to its center and is computed as $A^*_{i,a} = \frac{1}{|\mathcal{C}^*_{i,a}|}\sum_{x_a \in \mathcal{C}^*_{i,a}} d(x_a, \phi^*(x_a)) = L^*_{p,i,a}/|\mathcal{C}^*_{i,a}|$. Now consider r' be the first round when the

Algorithm 9: Fully online COCA$_F$

Input: set of datapoints X, protected group function $\rho : X \to [m]$ and capacity constraint $\boldsymbol{\gamma} = \{\gamma_a\}_{a=1}^m$

Output: cluster centers C and assignment function ϕ

Initialize $\Gamma \leftarrow \varnothing$ `// stores vacant capacity of center`
Open first H_k data points as centers (Claim 5.1) and $\forall j \in [H_k]$, $\forall a \in [m]$ set $\Gamma(c_j, a) = \gamma_a$
Initialize $\phi \leftarrow \varnothing$, $\forall a \in [m]$ set $\{r_a \leftarrow 1,\, n_{a,r_a} \leftarrow H_k,\, q_{a,r_a} \leftarrow H_k,\, idx_a \leftarrow H_k\}$
$\ell_p^* \leftarrow$ objective cost on H_k using Mulvey and Beck [242].
Initialize center opening cost $f_{a,r} = (\ell_p^* \vartheta_a)/(k \log n_{a,r_a})$ $\forall a \in [m]$
for *each remaining* $x_t \in X$ **do**
    $a \leftarrow \rho(x_t)$
    $c \leftarrow \operatorname{argmin}_{c \in C : \Gamma(c,a) > 0} d(x_t, c)$
    With probability $p_t = \min\left(1, d(x_t, c)/f_{a,r_a}\right)$:
        $C \leftarrow C \cup \{x_t\}$
        $\phi(x_t) = x_t$
        $\Gamma(x_t, a) = \gamma_a$
        $q_{a,r_a} \leftarrow q_{a,r_a} + 1$
    Otherwise, with probability $1 - p_t$:
        $\phi(x_t) = c$
        $\Gamma(c, a) = \Gamma(c, a) - 1$
    if $q_{a,r_a} \geq \frac{3k}{\vartheta_a}(1 + \log n_{a,r_a})$ **then**
        $r_a \leftarrow r_a + 1$
        $q_{a,r_a} \leftarrow 0$
        $f_{a,r_a} \leftarrow 2 f_{a,r_a - 1}$
    end
    $idx_a \leftarrow idx_a + 1$
    if $idx_a \geq n_{a,r_a}$ **then**
        $n_{a,r_a} \leftarrow 2 n_{a,r_a}$
    end
end
return (C, ϕ)

center opening cost $f_{a,r'}$ becomes some fraction of L_p^* such that, $f_{a,r'} \geq \frac{2^4 L_p^* \vartheta_a}{k \log n_a}$. Now, we bound the expected number of centers in two separate parts, i.e., before round r' and second during and after round r'. Let us first begin with the former,

Case 1: By the definition of r', we have $f_{a,r'-1} < \frac{2^4 L_p^* \vartheta_a}{k \log n_a}$. Further, since the center opening cost becomes twice at every round, we have $f_{a,r'-1} = 2^{r'-1} f_{a,1}$. Substituting the value of $f_{a,1} = \frac{\ell_p^* \vartheta_a}{k \log(n_a)}$, we get, $r' \leq \log\left(\frac{L_p^*}{\ell_p^*}\right) + 5$. Therefore, before round r', the number of centers opened by the algorithm is for group value $a \in [m]$,

$$\mathbb{E}(|C|^a_{\text{before } r'}) = O\left(\frac{3k}{\vartheta_a}(1 + \log n_a) \log\left(\frac{L_p^*}{\ell_p^*}\right)\right) \tag{5.6}$$

Thus, summing the result for all group values,

$$\mathbb{E}(|C|_{\text{before } r'}) = \sum_{a\in[m]} \mathbb{E}(|C|^a_{\text{before } r'})$$

Now, let $a \in [m]$ be the most dominant group value in the dataset that is $\exists a \in [m] : n_a \geq n_b \; \forall b \in [m]$ then as logarithmic function is a monotonically increasing function we have $\log n_a \geq \log n_b \; \forall b \in [m]$ and $n_a >> v_a$. Therefore,

$$\mathbb{E}(|C|_{\text{before } r'}) \leq \frac{3mk}{\vartheta_a}(1+\log n_a)\log\left(\frac{L^*_p}{\ell^*_p}\right) \tag{5.7}$$

Case 2: Now, let's look into computing the number of centers opened during and after round r' in each of these rings. To this, we again divide the bounds into three subparts-
Case 2(a): First, we estimate the number of new centers that will open for the first time in each ring for every group value. Let's denote these centers as K^1_τ. Since there are a total of $(1+\log n_a)$ rings in each cluster for each group value $a \in [m]$, therefore the total number of such centers are $\sum_k \sum_\tau \sum_a 1 = mk(1+\log n_a)$.
Case 2(b): Next, suppose there is a data point x that arrives and the closest center to x has already reached its capacity; in such a case, the data point will continue searching for the next closest center in any of the rings in increasing order of distance. There are two possibilities: either data point x will find a vacant center, or its likelihood of becoming a center increases as it delves further into the chain if the next closest center is too far away (handled in the next case). Let's denote the extra number of centers that need to be opened up in any ring $\boldsymbol{\tau}$ due to the exhausting capacity of the first centers of Case 2a, be denoted by K^c_τ. It is important to note that once a center K^1_τ fills up and we need to open a second center within the ring, then at least $\frac{\vartheta_a n_a}{k}$ data points have already arrived and been assigned. Therefore, the total number of such K^c_τ centers over all rings and clusters is upper bounded by k/ϑ_a i.e., $\sum K^c_\tau \leq \sum_{a\in[m]} k/\vartheta_a \leq \frac{mk}{\min_{a\in[m]}(\vartheta_a)}$.
Case 2(c): Now since there were two possibilities: either data point x will find a vacant center, or its likelihood of becoming a center increases as it delves further into the chain if the next closest center is too far away. Now, the remaining task is to bound the probabilistically opened centers in each ring apart from the centers opened in the previous two subcases. To do this according to Algorithm 9, if a data point x is the initial center opened within any ring, then the probability of subsequent point x' from the same ring opening as a center is defined and bounded using the properties of rings as follows:

$$\frac{d(x,x')}{f_{a,r'}} \leq \frac{d(x,\phi^*(x)) + d(x',\phi^*(x))}{f_{a,r'}} \leq 2\cdot 2^\tau \frac{A^*_i}{f_{a,r'}}$$
($\because$ using triangular inequality and ring property)

So, the expected number of centers that will open in any ring for all m group values over all rounds $r \geq r'$ is $\sum_{r\geq r'} \sum_{a=1}^m \frac{2\cdot 2^\tau A^*_i}{f_{a,r}} |\mathcal{C}^*_{i,\tau,r}|$. Summing these probabilistic centers over all rings and using $f_{a,r} \geq f_{a,r'}$ we obtain the number of centers opened for estimating one optimal cluster center as,

$$
\begin{aligned}
K^p_{\boldsymbol{\tau}} &= \sum_{\tau\geq 0}\left(\sum_{r\geq r'}\sum_{a=1}^{m}\frac{2\cdot 2^{\tau}A^*_{i,a}}{f_{a,r}}|\mathcal{C}^*_{i,\tau,r,a}|\right)\\
&\leq \sum_{\tau\geq 0}\sum_{a=1}^{m}\frac{2\cdot 2^{\tau}A^*_{i,a}}{f_{a,r'}}\sum_{r\geq r'}|\mathcal{C}^*_{i,\tau,r,a}|\\
&\leq \sum_{\tau\geq 0}2\cdot 2^{\tau}\sum_{a=1}^{m}\frac{A^*_{i,a}}{f_{a,r'}}|\mathcal{C}^*_{i,\tau,a}|\\
&\leq \sum_{a=1}^{m}\frac{2A^*_{i,a}|\mathcal{C}^*_{i,0}|}{f_{a,r'}}+\sum_{a=1}^{m}\frac{4}{f_{r',a}}\sum_{\tau\geq 1}\sum_{x_a\in\mathcal{C}^*_{i,\tau,a}}2^{\tau-1}A^*_{i,a}\\
&\leq \sum_{a=1}^{m}\frac{2A^*_{i,a}|\mathcal{C}^*_{i,0}|}{f_{a,r'}}+\sum_{a=1}^{m}\frac{4}{f_{r',a}}\sum_{\tau\geq 1}\sum_{x_a\in\mathcal{C}^*_{i,\tau,a}}d(x_a,\phi^*(x_a))\\
&\leq \sum_{a=1}^{m}\frac{2A^*_{i,a}|\mathcal{C}^*_{i,0}|}{f_{a,r'}}+\sum_{a=1}^{m}\frac{4L^*_{p,i,a}}{f_{r',a}}\\
&\leq \frac{6mL^*_{p,i}}{f_{b,r'}} && (\exists b\in[m]: f_{a,r'}\geq f_{b,r'}\ \forall a\in[m])\\
&\leq \frac{6mL^*_{p,i}}{f_{r',b}}
\end{aligned}
$$

(using $L^*_{p,i,a}=A^*_{i,a}|\mathcal{C}^*_{i,a}|=\sum_{x_a\in\mathcal{C}^*_{i,a}}d(x_a,\phi^*(x_a))$ and $2^{\tau-1}A^*_{i,a}\leq d(x_a,\phi^*(x_a))\ \forall x_a$)

Summing this up for all k cluster centers and considering the estimate of $f_{b,r'}\geq\frac{16L^*_p\vartheta_b}{k\log n_b}$ we get,

$$K^p_k\leq\frac{6mL^*_p}{f_{b,r'}}\leq\frac{6mk\log n_b}{16\vartheta_b}\tag{5.8}$$

Therefore, number of total centers opened in Case 2 are as follows

$$\mathbb{E}(|C|_{\text{during and after }r'})=O\left(K^1_{\boldsymbol{\tau}}+\sum_{\boldsymbol{\tau},k}K^c_{\boldsymbol{\tau}}+K^p_k\right)$$

$$=O\left(\frac{3mk}{\vartheta_a}(1+\log n_a)\log\left(\frac{L^*_p}{\ell^*_p}\right)+\frac{mk}{\min_{a\in[m]}(\vartheta_a)}+\frac{6mk\log n_b}{16\vartheta_b}\right)\tag{5.9}$$

where $a\in[m]$ is the most dominant group value in the data stream and $b\in[m]$ is the least dominant group value.

Now, since fully online algorithms are not aware of the values of L^*_p,ℓ^*_p so, following the lines as in Theorem 5.5, we consider, $\ell^*_p\geq\min_{x,x'\in X:x\neq x'}d(x,x')$. Similarly let, $L^*_p\leq n\max_{x,x'\in X}d(x,x')$. Substituting these values and adding initial H_k centers completes the proof. □

Theorem 5.8. *Let $L^{\texttt{COCA}_\texttt{F}}_p$ be the cost of online fair Algorithm 9 and L^*_p be the optimal offline capacitated cost. Then, $\mathbb{E}(L^{\texttt{COCA}_\texttt{F}}_p)=O\left(L^*_p\right)$.*

Proof. On the same lines as Theorem 5.3, 5.6, we bound the cost in two cases. In the

first case, we bound the cost using Lemma 5.4. We now consider R as the last round such that $f_{a,R} > f_{a,r'_a}$ for any round $r' < R$. Then case 1 can be loosely upper bounded as $O(mf_{a,R_a})$ where $a \in [m]$ is the group value of data points that are available in most abundance in the stream. The result uses the reasoning that the group most abundant in the stream will probably have the highest (or doubling) facility opening cost towards the last rounds or termination. The second case which dealt with bounding the data points that are allocated within the ring and getting the upper bound of L_p^*. To this, we have the total expected cost over all rings as $O(mf_{a,R_a}k\log n_a + L_p^*)$ where $a \in [m]$ is the most dominant group in terms of the number of total data points in the stream. Thus, we now need to estimate the value of f_{a,R_a} at the last round R. Let us consider any round r such that $f_{a,r_a} \geq \frac{16L_p^*\vartheta_a}{k\log(n_{a,r_a})}$. Here n_{a,r_a} are the number of data points that have been processed in online fashion from the data stream belonging to group value $a \in [m]$. Then, the number of centers opened in round r is given as $q_{a,r_a} \leq \frac{k}{\vartheta_a}(1+\log(n_{a,r_a})) + q'_{a,r_a}$. Here, we pessimistically count one (first) centers in each ring up to round r and q'_{a,r_a} is the number of centers opened in rings after opening former $\lceil k/\vartheta_a \rceil$ centers. In order to have more rounds than r, $\texttt{COCA}_F$ needs $q'_{a,r_a} \geq \frac{2k}{\vartheta_a}(1+\log(n_{a,r_a}))$. We will now compute the probability that $\texttt{COCA}_F$ terminates by round r. Applying Markov inequality by using the above information along with $\mathbb{E}(q'_{a,r_a}) \leq 6L_p^*/f_{a,r_a}$, we get the probability of reaching the next round as at most $3/16$. Thus, if b is the probability that $\texttt{COCA}_F$ terminates before round r. We have,

$$\begin{aligned}
\mathbb{E}(f_{a,R_a}) &= bf_{a,r_a-1} + (1-b)\sum_{r'=r}^{\infty} f_{a,r_a}\left(\frac{13}{16}\right)\left(\frac{3}{16}\right)^{r'-r} \\
&< bf_{a,r_a} + f_{a,r_a}(1-b)\left(\frac{13}{16}\right)\sum_{i=0}^{\infty} 2^i\left(\frac{3}{16}\right)^i \\
&= O(f_{a,r_a}) = O\left(\frac{16m\vartheta_a L_p^*}{k\log(n_{a,r_a})}\right). \qquad \text{(using } f_{a,r_a-1} = 2f_{a,r_a} \text{ and } b \leq 1)
\end{aligned}$$

On substituting this back, we get

$$\mathbb{E}(L_p^{\texttt{COCA}_\texttt{F}}) = O(mf_{a,R_a}k\log(n_a) + L_p^*) = O\left(\frac{16L_p^*\vartheta_a k\log n_a}{k\log n_{a,r_a}} + L_p^*\right)$$

such that $a \in [m]$ be the most dominant group in the stream. Now, since with high probability, the algorithm will terminate at r^{th} round and from doubling trick, we can say, $n_r \geq n$. So, $\mathbb{E}(L_p^{\texttt{COCA}_\texttt{F}}) = O\left(L_p^*\vartheta_a + L_p^*\right) = O(L_p^*)$. This completes the proof. Additionally, all results hold for any scalar distance metric.

□

5.7 Experimental Results and Discussion

We will now validate our proposed approaches against `SOTA` on following datasets motivated by clustering literature [44]:

- **Synthetic1d**: consist of 1000 points sampled each from $\{\mathcal{N}_i(\mu = 1+a\cdot i, \sigma = 2)\}_{i=1}^{k}$, where $a = 7$ in well separable (s) and $a = 5$ in partially overlapping (o) clusters.

- **Synthetic2d**: consist of 1000 points sampled each from $\{\mathcal{N}_i(\mu = 1 + a \cdot i, \Sigma = I_{2\times 2})\}_{i=1}^{k}$, where variable '$a$' as above and I is identity matrix. Synthetic2d is shortened to Syn2d.

- **Adult**[5]: The data is collected from 32562 people comprising 21790 males and 10771 females during US Census 1994. The five feature attributes chosen for the present study are: age, fnlwgt, education_num, capital_gain, hours_per_week; and align with prior literature on clustering [44].

- **Bank**[6]: Marketing records of 411109 Portuguese campaigns. The dataset consists of 11568 samples from singles, 24928 from married, and 4612 from divorced people. The six feature attributes consistent with previous research [44] are – age, duration, campaign, cons.price.idx, euribor3m, nr.employed.

- **Diabetes**[7]: Medical records with $100,000$ instances collected over the last ten years from 130 US hospitals. It is collected from 54708 male and 47055 females. The feature attributes are age, time_in_hospital [44].

Experimental Setup: All experiments are performed on Intel Xeon with 280GB RAM, and Python 3.6. We report mean and standard deviation over ten independent runs and seed from set $\{0, 100, \ldots, 900\}$. Notably, the capacity parameter (ϑ) is such that $\vartheta \geq 1$, with $\vartheta = 1$ representing the most restrictive scenario, i.e., having uniform capacities. The code[8] is publicly available for use.
We divide the experimental analysis into two subsections. We first validate the efficacy of an unfair online capacitated clustering algorithm and then later investigate the fair variation of the method. We evaluate the performance of `COCA` against the following:

- **Uncapacitated Online k-means** We call fully online algorithm as `LIB` for comparison with `COCA` (see Section 5.3 for working of algorithm). A heuristic approach is also provided by the authors in which they initially open $(k + 1)$ data points as centers and compute ℓ_p^*, by taking half sum of ten closest neighbours instead of the pairwise minimum distance between $(k + 1)$ data points. Further, they drop logarithmic factors by setting $q_r \geq k$ and increasing f_r by ten times instead of doubling it. We denote this as $\texttt{LIB}_\texttt{H}$ [38]. Note that while $\texttt{LIB}_\texttt{H}$ outperforms `LIB` but it lacks theoretical results to support it.

- **Capacitated Online Clustering Heuristic** (`COCH`): Motivated from $\texttt{LIB}_\texttt{H}$, we also use heuristic with a selection of $(k + 1)$ initial points as centers, setting $q_r \geq k$

[5] archive.ics.uci.edu/ml/datasets/Adult
[6] archive.ics.uci.edu/ml/datasets/Bank+Marketing
[7] archive.ics.uci.edu/dataset/116/us+census+data+1990
[8] https://github.com/shivi98g/Capacitated-Online-Clustering-Algorithm

and updating f_r by ten times in COCA to enable comparison of COCH with $\texttt{LIB}_\texttt{H}$ (pseudo-code in Algorithm 10).

- **Offline Capacitated Clustering** (CAP): Assigns data points to closest vacant centers and performs mean (or median) center updates as in heuristic [242]. We compare the algorithm with both k-means and k-median variation of the method.

Additionally, for comparison of $\texttt{COCA}_F$, we introduce an additional heuristic method similar to $\texttt{LIB}_\texttt{H}$, COCH. We call it $\texttt{COCH}_F$. It uses a heuristic with a selection of $(k+1)$ initial points as centers, setting $q_r \geq k$ and updating f_r by ten times in $\texttt{COCA}_F$. We now discuss the metrics on which we compare these methods.

Metric: We re-introduce: $k_{\texttt{target}}$, $k_{\texttt{actual}}$. The former is the input for any online algorithm, while the latter represents the count of final centers opened. Also, we compare COCA's performance on cost. It involves comparing the cost of online solutions when executed for $k_{\texttt{target}}$ as input (resulting in $k_{\texttt{actual}}$ centers) v/s their offline counterparts when executed and compared on $k_{\texttt{actual}}$ centers. **An important note** that since LIB and the proposed COCA have theoretical guarantees and should thus be compared. In contrast, $\texttt{LIB}_\texttt{H}$ and proposed COCH are heuristic approaches that warrant comparison.

5.7.1 Analysis in Unfair Online Setting

The first half of the experimental analysis focuses on the online capacitated setting with the capacity constraint on overall cluster sizes, i.e., we compare proposed COCA, COCH with the uncapacitated online clustering methods. We will later see in Section 5.7.2 the fair setting where capacity constraints are provided for each protected group at every cluster.

Analysis on Number of Centers Opened

–**Under uniform capacities** ($\vartheta = 1$): We compare the centers opened by COCA, COCH with SOTA. Results for on $k_{\texttt{target}}$ of $2, 3, 5, 7, 10, 15, 20, 25, 30$ and 40 are listed in Tables 5.1 to 5.10. As can be observed from the tables, though the number of centers opened for the lower value of target k are slightly larger in a few datasets for COCA as compared to LIB due to capacity constraints. But as the value of target k increases, we see that COCA results in significantly fewer cluster centres than LIB. It is primarily due to our idea of opening H_k number of initial centers instead of opening only $(k+1)$ points. This helps in getting a better estimate of the lower bound on optimal cost. Further, we can observe that the proposed COCA and COCH demonstrate significantly lower deviations than LIB, $\texttt{LIB}_\texttt{H}$. This is attributed to the doubling trick instead of increasing the estimate of the number of points in a linear fashion. Reduced variance helps online algorithms avoid opening more centers than the target. For heuristics, the gap between $k_{\texttt{target}}$, $k_{\texttt{actual}}$ is tolerable for lower targets and slightly high in higher targets, considering the rising uncertainty of the arrival order of points.

Algorithm 10: Fully online `COCH`

Input: stream X and capacity parameter γ
Output: cluster centers C and assignment function $\hat{\phi}$

Initialize $\Gamma \leftarrow \varnothing$ `// stores vacant capacity of center`
Open first $k+1$ points as centers and $\forall j \in [k+1]$ set $\Gamma(c_j) = \gamma$
Initialize $\phi \leftarrow \varnothing$, $r \leftarrow 1$, $q_r \leftarrow k+1$
$\ell_p^* \leftarrow$ objective cost on $k+1$ points using Mulvey and Beck [242].
Initialize center opening cost $f_r = \ell_p^* \vartheta / k$
for *each remaining* $x_t \in X$ **do**
  $c \leftarrow \text{argmin}_{c \in C : \Gamma(c) > 0} \, d(x_t, c)$
  With probability $p_t = \min\left(d(x_t, c)/f_r, 1\right)$:
    $C \leftarrow C \cup \{x_t\}$
    $\phi(x_t) = x_t$
    $\Gamma(x_t) = \gamma$
    $q_r \leftarrow q_r + 1$
  Otherwise, with probability $1 - p_t$:
    $\phi(x_t) = c; \quad \Gamma(c) = \Gamma(c) - 1.$
  if $q_r \geq k$ **then**
    $r \leftarrow r + 1; \quad q_r \leftarrow 0; \quad f_r \leftarrow 10 f_{r-1}.$
  end
end
return (C, ϕ)

–**Under relaxed capacities** ($\vartheta > 1$): We also evaluate the performance of `COCA` and `COCH` as the capacity parameter (ϑ) is relaxed from 1 to $2.5, 5, 7$, and $\texttt{k}_{\texttt{target}}$. The results are presented in Tables 5.11 to 5.15 for $\texttt{k}_{\texttt{target}}$ values spanning from 2 to 40. Remarkably, we observe that the number of opened centers remains relatively consistent even as we relax the constraint on ϑ from the most restrictive setting of one. This can be attributed to the fact that the online algorithms, whether capacitated or uncapacitated, open a sufficient number of centers to accommodate all points due to the unknown order of arrival of data points in the online setting. While there are a few datasets and target values where a slight difference (increase or decrease) in centers occurs.

Dataset	`LIB`	`COCA`	$\texttt{LIB}_{\texttt{H}}$	`COCH`
Adult	292.9±20.79	541.7 ± 79.80	9.0±0.0	9.0±0.0
Bank	311.9±33.29	544.3 ± 82.69	9.0±0.0	9.0±0.0
Diabetes	109.5±15.18	143.5 ± 16.55	8.6±0.79	9.0±0.0
Syn2d-(s)	134.2±45.52	113.9 ± 19.32	7.0±1.54	8.5±0.67
Syn2d-(o)	142.1±72.06	137.1 ± 21.39	7.0±0.44	7.8±0.60
Syn1d-(s)	104.8±98.75	57.6 ± 8.18	6.8±1.32	6.2±0.75
Syn1d-(o)	66.6±34.67	61.9 ± 7.54	7.8±1.32	6.6±0.49

Table 5.1: Comparison against unfair `COCA`, `COCH` on $k_{\texttt{actual}}$ against various `SOTA` methods when $k_{\texttt{target}}$ is 2 under uniform capacities (i.e., ϑ is 1).

Dataset	LIB	COCA	LIB_{H}	COCH
Adult	474.0±67.56	766.4 ± 107.53	12.8±0.6	13.8±0.87
Bank	522.1±71.14	737.5 ± 98.59	12.1±1.13	13.4±0.79
Diabetes	145.4±22.64	141.0 ± 13.59	12.0±1.09	12.8±0.6
Syn2d-(s)	197.5±56.34	212.9 ± 31.09	8.9±1.37	10.1±0.3
Syn2d-(o)	197.5±56.34	197.0 ± 25.82	9.3±1.01	10.3±0.45
Syn1d-(s)	137.7±108.37	82.2 ± 8.67	9.9±1.3	10.0±0
Syn1d-(o)	142.4±92.71	77.4 ± 8.94	9.8±1.24	10.1±0.3

Table 5.2: Comparison against unfair COCA, COCH on $k_{\texttt{actual}}$ against various SOTA methods when $k_{\texttt{target}}$ is 3 under uniform capacities (i.e., ϑ is 1).

Dataset	LIB	COCA	LIB_{H}	COCH
Adult	806.7±93.32	1012.1 ± 133.66	20.0±1.34	21.7±1.18
Bank	886.1±115.89	1095.7 ± 133.58	19.7±1.48	21.4±1.2
Diabetes	195.8±26.97	170.5 ± 12.75	17.5±0.80	19.1±1.51
Syn2d-(s)	454.9±173.15	302.8 ± 35.08	12.9±1.3	16.0±1.48
Syn2d-(o)	454.9±173.15	339.8 ± 33.11	13.6±1.2	16.9±1.3
Syn1d-(s)	263.9±105.39	103.6 ± 18.34	13.4±1.68	17.9±3.38
Syn1d-(o)	241.9±88.46	102.7 ± 19.19	13.7±1.61	17.7±2.93

Table 5.3: Comparison against unfair COCA, COCH on $k_{\texttt{actual}}$ against various SOTA methods when $k_{\texttt{target}}$ is 5 under uniform capacities (i.e., ϑ is 1).

Dataset	LIB	COCA	LIB_{H}	COCH
Adult	1153.4±159.58	1220.5 ± 126.19	26.3±1.73	29.3±0.45
Bank	1311.6±137.23	1388.1 ± 180.04	26.4±2.01	29.4±0.66
Diabetes	214.4±25.81	163.8 ± 8.07	22.3±0.64	24.9±2.11
Syn2d-(s)	826.6±166.96	362.2 ± 41.35	17.9±2.62	23.4±3.50
Syn2d-(o)	826.6±166.96	390.5 ± 60.86	19.5±2.20	23.6± 2.53
Syn1d-(s)	401.1±168.74	123.2 ± 12.79	18.1±2.94	21.9±1.57
Syn1d-(o)	431.3±190.35	127.5 ± 17.51	18.4±2.61	22.0±1.18

Table 5.4: Comparison against unfair COCA, COCH on $k_{\texttt{actual}}$ against various SOTA methods when $k_{\texttt{target}}$ is 7 under uniform capacities (i.e., ϑ is 1).

Dataset	LIB	COCA	LIB_{H}	COCH
Adult	1857.4±206.12	1735.8 ± 202.95	36.5±2.57	41.0±0.44
Bank	1969.5±205.36	1751.4 ± 191.69	37.3±3.00	41.3±0.64
Diabetes	246.3±20.34	189.0 ± 16.05	31.1±0.3	33.9±2.7
Syn2d-(s)	1357.8±358.48	619.3 ± 90.66	29.7±4.10	32.5±3.90
Syn2d-(o)	1357.8±358.48	675.6 ± 99.39	31.0±3.34	33.8±4.77
Syn1d-(s)	938.4±560.99	252.1 ± 44.50	28.7±2.9	31.0±0.44
Syn1d-(o)	910.4±485.25	239.1 ± 34.03	28.7±2.45	31.1±0.3

Table 5.5: Comparison against unfair COCA, COCH on $k_{\texttt{actual}}$ against various SOTA methods when $k_{\texttt{target}}$ is 10 under uniform capacities (i.e., ϑ is 1).

Dataset	LIB	COCA	LIB_H	COCH
Adult	3143.1±866.86	2205.6 ± 251.65	54.5±5.46	60.9±0.3
Bank	3273.6±504.60	2325.3 ± 246.43	59.2±2.56	63.9±2.91
Diabetes	265.8±4.70	206.0 ± 7.94	50.7±5.86	47.9±2.80
Syn2d-(s)	2240.3±379.80	948.0 ± 126.29	51.6±5.14	57.4±5.00
Syn2d-(o)	2240.3±379.80	1031.8 ± 127.52	51.9±4.88	57.8±3.40
Syn1d-(s)	1536.4±904.62	292.8 ± 62.04	47.8±2.22	51.7±7.75
Syn1d-(o)	1603.2±1149.23	331.3 ± 71.18	46.9±2.54	48.3±2.83

Table 5.6: Comparison against unfair COCA, COCH on $k_{\texttt{actual}}$ against SOTA methods when $k_{\texttt{target}}$ is 15 under uniform capacities (i.e., ϑ is 1).

Dataset	LIB	COCA	LIB_H	COCH
Adult	4657.4±1060.78	2579.4 ± 258.59	77.1±5.37	81.0±0.0
Bank	4469.8±544.19	2867.3 ± 309.31	80.3±2.09	82.4±2.11
Diabetes	268.8±1.4	211.3 ± 15.58	71.5±7.81	66.4.1±6.74
Syn2d-(s)	3713.8±750.72	964.7 ± 110.01	65.7±4.63	62.6±6.29
Syn2d-(o)	3713.8±750.72	1175.0 ± 177.38	72.3±6.45	68.5±6.21
Syn1d-(s)	2136.1±674.30	451.1 ± 44.47	66.6±6.37	69.0±5.13
Syn1d-(o)	2141.2±432.02	498.7 ± 56.46	69.3±7.03	80.3±2.23

Table 5.7: Comparison against unfair COCA, COCH on $k_{\texttt{actual}}$ against SOTA methods when $k_{\texttt{target}}$ is 20 under uniform capacities (i.e., ϑ is 1).

Dataset	LIB	COCA	LIB_H	COCH
Adult	5946.6±1382.66	2858.7 ± 284.03	94.7±7.31	100.3±2.79
Bank	5571.2±701.10	3007.6 ± 256.45	98.6±4.94	102.1±1.04
Diabetes	270.1±1.3	224.4 ± 9.44	124.9±50.44	83.6±6.96
Syn2d-(s)	4939.5±1032.33	1117.0 ± 98.45	88.9±6.87	88.9±7.59
Syn2d-(o)	4939.5±1032.33	1365.3 ± 153.64	93.3±8.05	95.3±6.35
Syn1d-(s)	3565.6±1281.58	407.2 ± 55.26	88.9±6.87	72.9±7.32
Syn1d-(o)	3722.4±1478.66	443.2 ± 70.21	92.4±8.69	75.5±6.75

Table 5.8: Comparison against unfair COCA, COCH on $k_{\texttt{actual}}$ against SOTA methods when $k_{\texttt{target}}$ is 25 under uniform capacities (i.e., ϑ is 1).

Dataset	LIB	COCA	LIB_H	COCH
Adult	7136.4±1617.56	3071.1 ± 251.07	114.7±8.1	118.7±4.00
Bank	6756.1±834.40	3485.0 ± 335.15	116.2±6.24	122.9 ± 1.58
Diabetes	271.5±1.74	252.7 ± 12.46	198.7±75.14	93.6±2.42
Syn2d-(s)	6728.9±1585.75	1479.2 ± 201.29	116.8±7.39	94.2±3.89
Syn2d-(o)	6728.9±1585.75	1775.5 ± 222.62	120.2±5.05	97.0±4.63
Syn1d-(s)	4489.2±1692.76	601.4 ± 50.25	116.8±7.39	91.9±0.83
Syn1d-(o)	4055.8±1609.87	567.1 ± 44.78	112.5±8.64	92.5±1.69

Table 5.9: Comparison against unfair COCA, COCH on $k_{\texttt{actual}}$ against various SOTA methods when $k_{\texttt{target}}$ is 30 under uniform capacities (i.e., ϑ is 1).

Dataset	LIB	COCA	LIB_{H}	COCH
Adult	9865.8±1889.41	3594.8 ± 294.33	154.4±8.19	159.6±3.93
Bank	9516.1±1094.63	3744.7 ± 576.77	161.3±2.53	162.1±1.92
Diabetes	274.4 ± 2.53	323.3 ± 7.87	251.8±47.46	117.0±9.04
Syn2d-(s)	8872.2±1635.30	1756.2 ± 143.67	155.2±9.17	138.9±18.07
Syn2d-(o)	8872.2±1638.30	2134.1 ± 238.14	156.9±12.41	145.6±13.74
Syn1d-(s)	5744.6±1227.76	786.5 ± 74.76	155.2±9.17	117.1±16.94
Syn1d-(o)	5955.8±1470.21	801.0 ± 70.28	155.4±10.73	118.4±20.34

Table 5.10: Comparison against unfair COCA, COCH on $k_{\texttt{actual}}$ against SOTA methods when $k_{\texttt{target}}$ is 40 under uniform capacities (i.e., ϑ is 1).

Dataset	$\vartheta = 1$	$\vartheta = 2.5$	$\vartheta = 5$	$\vartheta = 7$	$\vartheta = \text{k}_{\texttt{target}}$
Adult	541.7 ± 79.80	498.0 ± 73.71	515.5 ± 79.13	533.1 ± 82.67	578.9 ± 93.62
Bank	544.3 ± 82.69	504.2 ± 76.26	545.3 ± 84.54	533.1 ± 82.67	509.4 ± 78.28
Diabetes	143.5 ± 16.55	152.6 ± 18.67	127.0 ± 15.47	137.7 ± 17.06	131.7 ± 17.92

Table 5.11: $k_{\texttt{actual}}$ on COCA methods when $k_{\texttt{target}}$ is 2 under varying capacity parameter (i.e., ϑ values).

Dataset	$\vartheta = 1$	$\vartheta = 2.5$	$\vartheta = 5$	$\vartheta = 7$	$\vartheta = \text{k}_{\texttt{target}}$
Adult	766.4 ± 107.53	766.4 ± 107.53	766.4 ± 107.53	766.4 ± 107.53	763.2 ± 103.11
Bank	737.5 ± 98.59	737.5 ± 98.59	737.5 ± 98.59	736.5 ± 98.98	736.2 ± 98.98
Diabetes	141.0 ± 13.59	141.0 ± 13.59	141.0 ± 13.59	141.0 ± 13.59	142.3 ± 13.50

Table 5.12: $k_{\texttt{actual}}$ on COCA methods when $k_{\texttt{target}}$ is 3 under varying capacity parameter (i.e., ϑ values).

Dataset	$\vartheta = 1$	$\vartheta = 2.5$	$\vartheta = 5$	$\vartheta = 7$	$\vartheta = \text{k}_{\texttt{target}}$
Adult	1735.80 ± 202.95	1735.80 ± 202.95	1735.80 ± 202.95	1735.80 ± 202.95	1746.1 ± 201.22
Bank	1751.4 ± 191.69	1751.4 ± 191.69	1751.4 ± 191.69	1751.4 ± 191.69	1748.7 ± 187.29
Diabetes	189.0 ± 16.05	189.0 ± 16.05	189.0 ± 16.05	189.0 ± 16.05	187.4 ± 15.04

Table 5.13: $k_{\texttt{actual}}$ on COCA methods when $k_{\texttt{target}}$ is 10 under varying capacity parameter (i.e., ϑ values).

Dataset	$\vartheta = 1$	$\vartheta = 2.5$	$\vartheta = 5$	$\vartheta = 7$	$\vartheta = \text{k}_{\texttt{target}}$
Adult	2858.7 ± 284.03	2858.7 ± 284.03	2858.7 ± 284.03	2858.7 ± 284.03	2857.4 ± 275.42
Bank	3007.6 ± 256.45	3007.6 ± 256.45	3007.6 ± 256.45	3007.6 ± 256.45	3002.1 ± 247.06
Diabetes	224.4 ± 9.44	224.4 ± 9.44	224.4 ± 9.44	224.4 ± 9.44	226.1 ± 8.45

Table 5.14: $k_{\texttt{actual}}$ on COCA methods when $k_{\texttt{target}}$ is 25 under varying capacity parameter (i.e., ϑ values).

Analysis on Clustering Cost

–**Comparison to k-means clustering**: We further validate our theoretical findings on the constant approximation of different online methods to their offline counterparts.

Dataset	$\vartheta = 1$	$\vartheta = 2.5$	$\vartheta = 5$	$\vartheta = 7$	$\vartheta = \mathtt{k_{target}}$
Adult	3594.80 ± 294.33	3594.80 ± 294.33	3594.80 ± 294.33	3594.80 ± 294.33	3592.0 ± 298.70
Bank	3744.7 ± 576.77	3744.7 ± 576.77	3744.7 ± 576.77	3744.7 ± 576.77	3907.9 ± 6347.39
Diabetes	323.3 ± 7.87	323.3 ± 7.87	323.3 ± 7.87	323.3 ± 7.87	324.2 ± 6.76

Table 5.15: $k_{\mathtt{actual}}$ on COCA methods when $k_{\mathtt{target}}$ is 40 under varying capacity parameter (i.e., ϑ values).

To this, we compare our fully online algorithm (COCA) to its offline capacitated k-means ($\mathtt{CAP}_{kms}$) (Figure 5.2). We further reproduce the result from uncapacitated online LIB to offline setting (k-means) (Figure 5.2). The Figure shows that our approximation factor ratio is near one for Adult and Bank datasets. For the Diabetes dataset, due to the existence of local minima [44] and the fact that sometimes the offline algorithm gets stuck in local optima, we observe that the ratio is slightly below one. We further see that for the most number of the values of $k_{\mathtt{target}}$, the ratio in the capacitated setting is lower than that of the uncapacitated setting. Further, we experimentally analyze the cost approximation factors of heuristic approaches in both capacitated and uncapacitated settings in Figure 5.3. Additionally, as an extended study, we even check how far is our cost approximation from online capacitated clustering COCA to offline uncapacitated k-means (standard vanilla algorithm) in Figure 5.4 and obtain similar constant cost approximation observations.

–**Comparison to k-median clustering**: We also replicate all cost comparison experiments using the k-median update objective. In this context, we substitute ($\mathtt{CAP}_{kms}$) with the offline capacitated k-median version ($\mathtt{CAP}_{kmd}$) introduced by Mulvey and Beck [242], and we replace offline uncapacitated k-means with offline uncapacitated k-median. The cost comparisons for COCA and LIB are presented against their capacitated and uncapacitated counterparts in Figures 5.5 and 5.6. Similarly, for COCH and, we present the results in Figures 5.7 and 5.8. In summary, we note constant cost approximations, mirroring k-means setting.

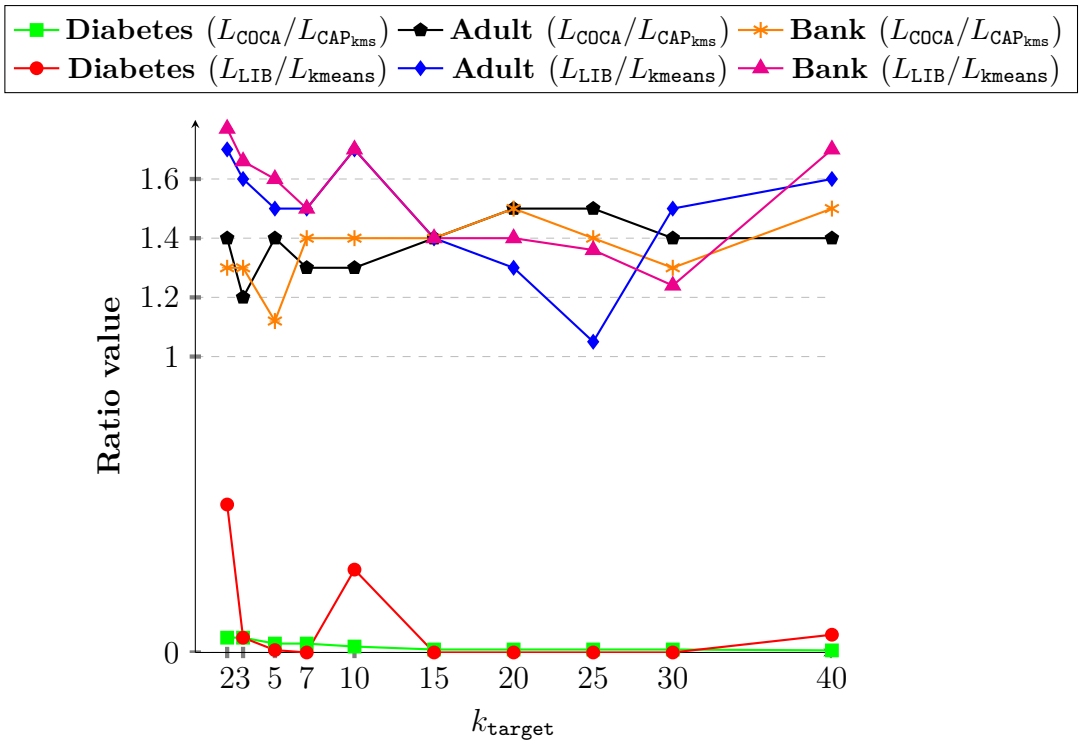


Figure 5.2: Cost approximation of COCA to offline capacitated k-means clustering ($\texttt{CAP}_{\texttt{kms}}$). Additionally, provide cost approximation of uncapacitated online clustering heuristic LIB to uncapacitated offline k-means.

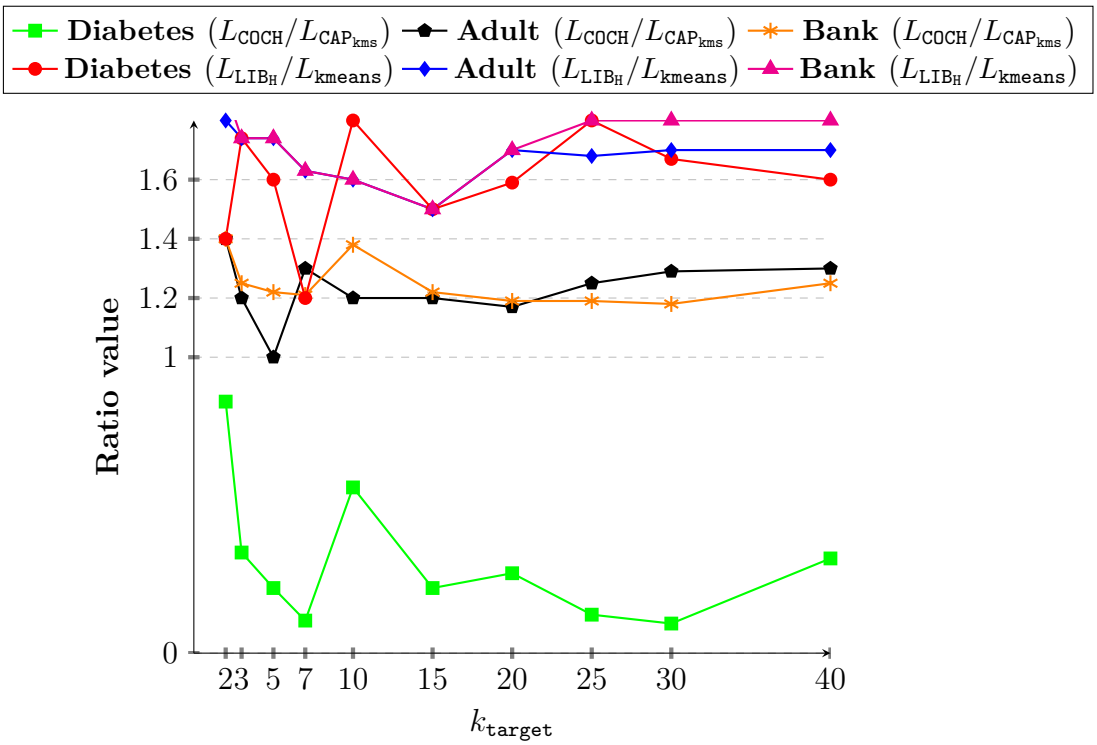


Figure 5.3: Cost approximation of COCH to offline capacitated clustering $\texttt{CAP}_{\texttt{kms}}$. Additionally, provide cost approximation of uncapacitated online clustering $\texttt{LIB}_{\texttt{H}}$ to uncapacitated offline k-means.

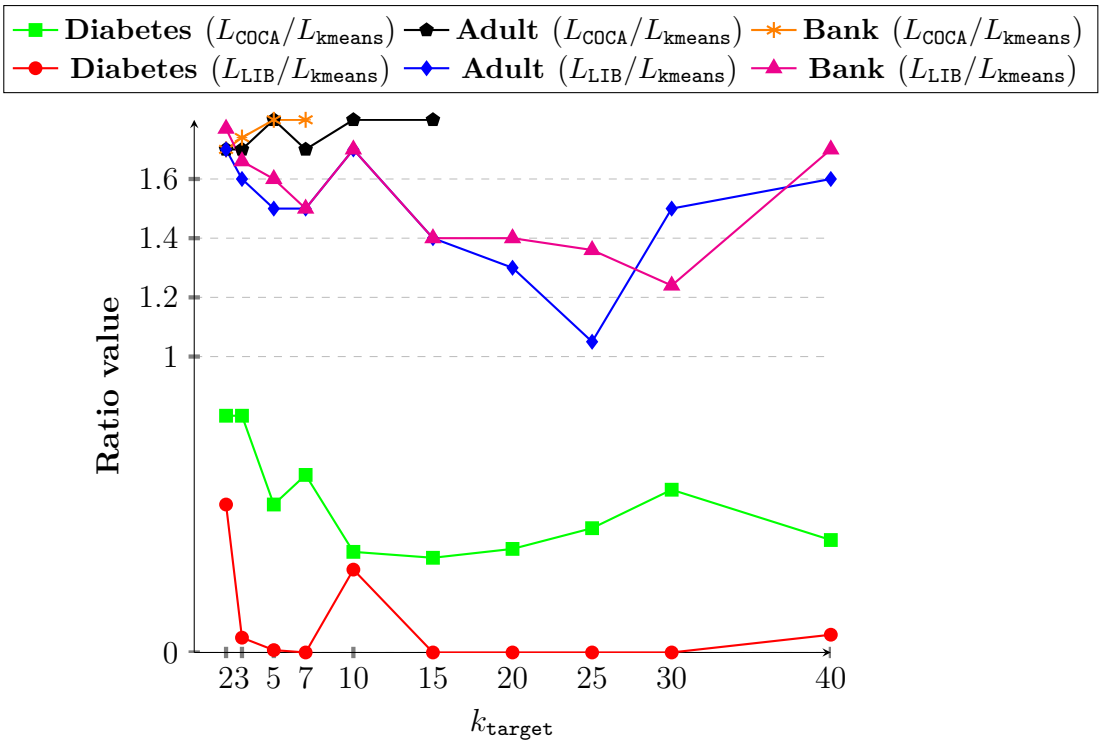


Figure 5.4: Cost approximation of `COCA` to offline capacitated clustering k-means. Additionally, provide cost approximation of uncapacitated online clustering `LIB` to uncapacitated offline k-means.

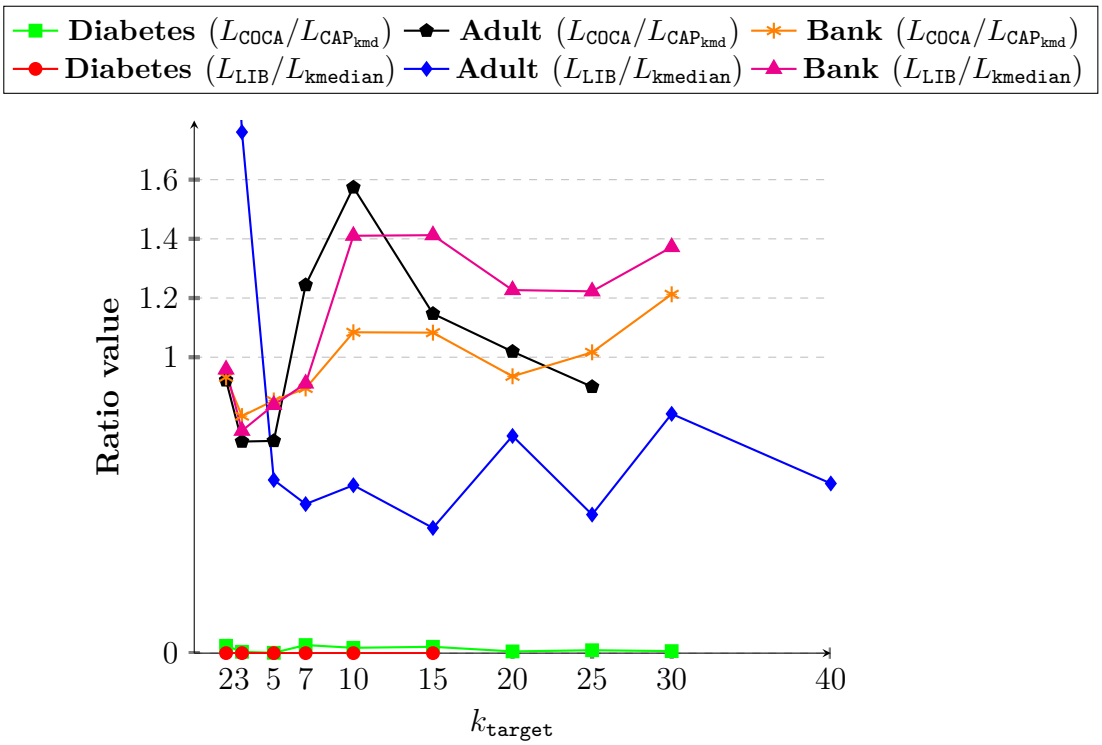


Figure 5.5: Cost approximation of `COCA` to offline capacitated k-median ($\texttt{CAP}_{\texttt{kmd}}$). Additionally, provide the level of comparison of cost approximation of uncapacitated online clustering heuristic `LIB` to uncapacitated offline k-median.

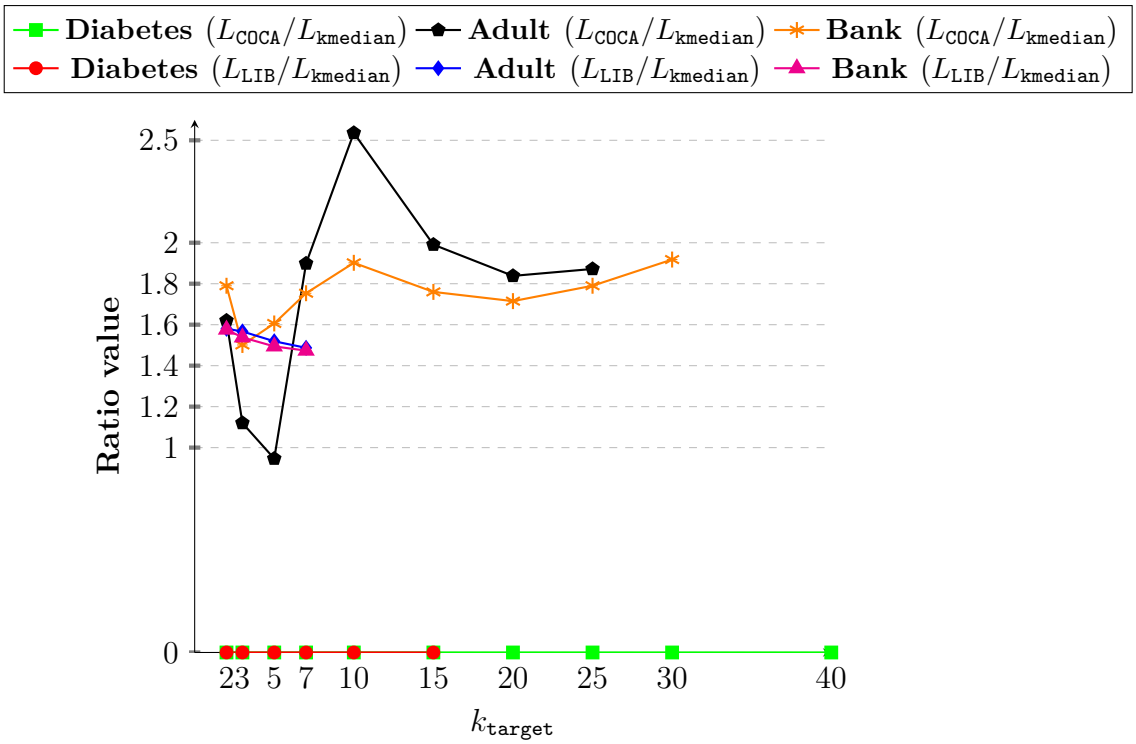


Figure 5.6: Cost approximation of `COCA` to offline capacitated clustering k-median. Additionally, provide cost approximation of uncapacitated online clustering `LIB` to uncapacitated offline clustering k-median.

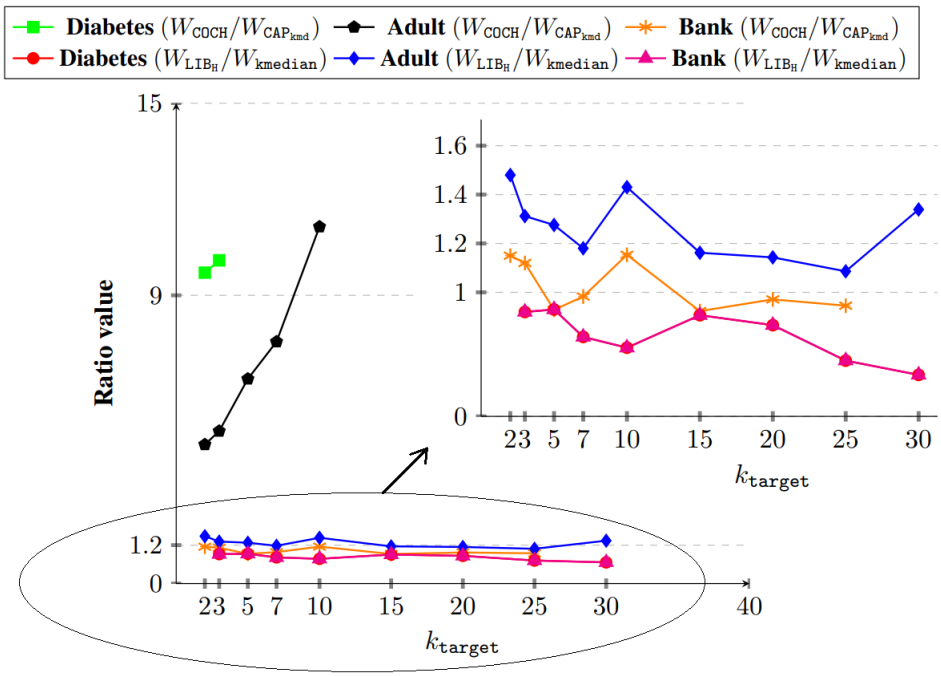


Figure 5.7: Cost approximation of `COCH` to offline capacitated k-median $\texttt{CAP}_{\texttt{kmd}}$. Additionally, provide cost approximation of uncapacitated online clustering $\texttt{LIB}_{\texttt{H}}$ to uncapacitated offline k-median.

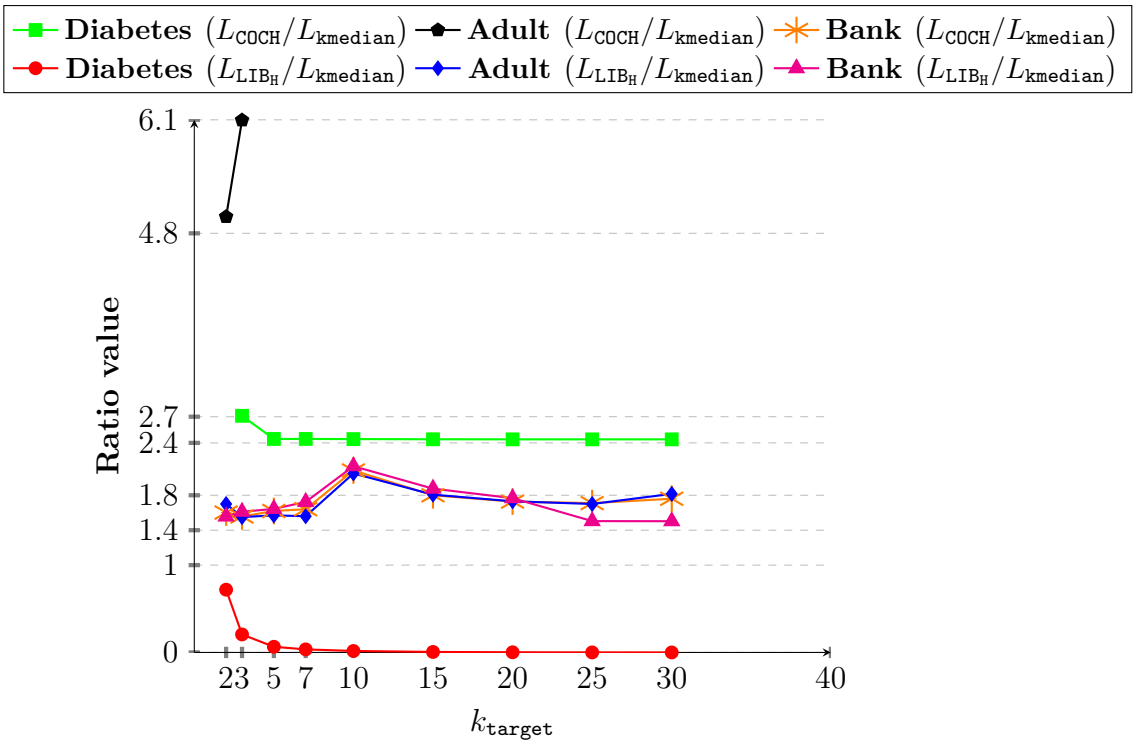


Figure 5.8: Comparison of cost approximation of COCH to offline capacitated clustering k-median. Additionally, the cost approximation of uncapacitated online clustering $\texttt{LIB}_\texttt{H}$ to uncapacitated offline clustering k-median is provided.

Ablation Study on Cluster Sizes

We now conduct an ablation study on cluster sizes for proposed COCA and LIB for uniform capacity and uncapacitated setting. The mean value of variance in cluster sizes across ten independent runs are reported in Table 5.16 to 5.20 for uniform and uncapacitated settings, respectively. The lower mean value indicates that across different runs, the variation in cluster sizes across all opened centers is not much and favourable in many instances, as discussed in the motivational example in the main paper. Note that in a uniform setting, we report results only for COCA as LIB is an online uncapacitated method and does not inherently accommodate capacity constraints. In an uncapacitated setting, it can be observed that the mean value is low especially on more challenging smaller $\texttt{k}_{\texttt{target}}$ and perform comparably to LIB as $\texttt{k}_{\texttt{target}}$ increases. Note that the COCA offers flexibility by allowing users to choose capacity constraints as per the necessity of real-world application.

	$\vartheta = 1$ (Uniform Capacity)		$\vartheta = k_{\texttt{target}}$ (Uncapacitated)	
Dataset	LIB	COCA	LIB	COCA
Adult	-	49.72	80.61	50.73
Bank	-	62.71	111.29	59.64
Diabetes	-	616.45	730.16	644.57

Table 5.16: The table reports the mean value of deviation in cluster sizes across ten independent runs on opening $k_{\texttt{actual}}$ clusters when $k_{\texttt{target}}$ is 2 for COCA, LIB. Since LIB is uncapacitated approach we report its result under $\vartheta = k_{\texttt{target}}$ setting.

	$\vartheta = 1$ (Uniform Capacity)		$\vartheta = k_{\texttt{target}}$ (Uncapacitated)	
Dataset	LIB	COCA	LIB	COCA
Adult	-	34.92	55.76	34.58
Bank	-	46.18	65.47	46.23
Diabetes	-	631.50	626.80	622.09

Table 5.17: The table reports the mean value of deviation in cluster sizes across ten independent runs on opening $k_{\texttt{actual}}$ clusters when $k_{\texttt{target}}$ is 3 for COCA, LIB. Since LIB is uncapacitated approach we report its result under $\vartheta = k_{\texttt{target}}$ setting.

	$\vartheta = 1$ (Uniform Capacity)		$\vartheta = k_{\texttt{target}}$ (Uncapacitated)	
Dataset	LIB	COCA	LIB	COCA
Adult	-	17.09	15.25	17.07
Bank	-	20.86	17.07	20.50
Diabetes	-	586.35	542.34	582.80

Table 5.18: The table reports the mean value of deviation in cluster sizes across ten independent runs on opening $k_{\texttt{actual}}$ clusters when $k_{\texttt{target}}$ is 10 for COCA, LIB. Since LIB is uncapacitated approach we report its result under $\vartheta = k_{\texttt{target}}$ setting.

	$\vartheta = 1$ (Uniform Capacity)		$\vartheta = k_{\texttt{target}}$ (Uncapacitated)	
Dataset	LIB	COCA	LIB	COCA
Adult	-	11.24	4.87	11.14
Bank	-	12.50	5.94	12.51
Diabetes	-	565.06	528.23	564.54

Table 5.19: The table reports the mean value of deviation in cluster sizes across ten independent runs on opening $k_{\texttt{actual}}$ clusters when $k_{\texttt{target}}$ is 25 for COCA, LIB. Since LIB is uncapacitated approach we report its result under $\vartheta = k_{\texttt{target}}$ setting.

	$\vartheta = 1$ (Uniform Capacity)		$\vartheta = k_{\texttt{target}}$ (Uncapacitated)	
Dataset	LIB	COCA	LIB	COCA
Adult	-	9.13	2.76	9.13
Bank	-	9.96	4.9006	9.97
Diabetes	-	508.84	526.12	508.78

Table 5.20: The table reports the mean value of deviation in cluster sizes across ten independent runs on opening $k_{\texttt{actual}}$ clusters when $k_{\texttt{target}}$ is 40 for COCA, LIB. Since LIB is uncapacitated approach we report its result under $\vartheta = k_{\texttt{target}}$ setting.

Reduction to Uncapacitated Problem (Unrestricted Setting)

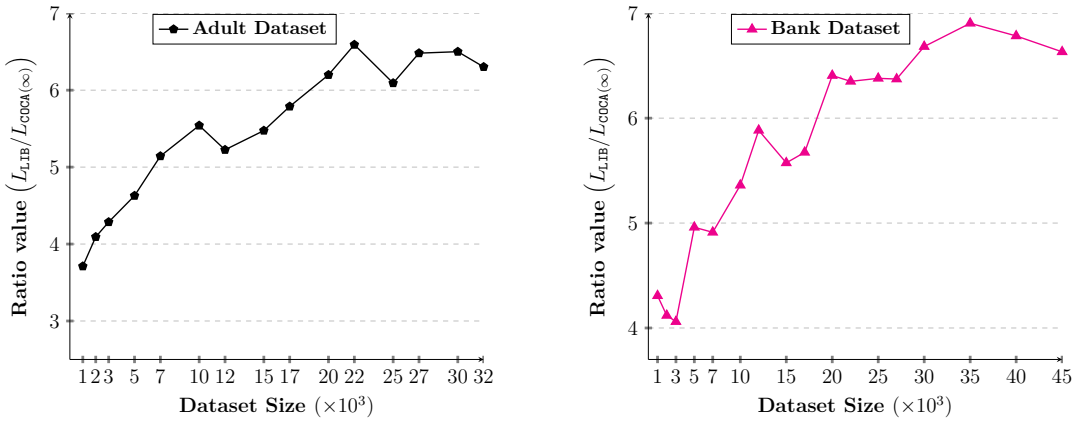
We set $\vartheta = k$ (unrestricted) in our algorithms and assess their performance on $k_{\texttt{actual}}$ and cost. The findings are summarized below:

–**Observation on Number of Centers Opened**: We report the comparison between uncapacitated COCA and LIB on the number of centers for different $\texttt{k}_{\texttt{target}}$ values in Table 5.21 to 5.30. The results on the number of centers resemble the uniform capacities but significantly better than LIB. Our observations indicate that, while for smaller target values LIB exhibits a slightly lower count of opened centers, the performance of LIB deteriorates as the target increases. This supports our choice to choose H_k as the initial set of centers compared to $k+1$ points in LIB.

–**Observation on cost**: Here, we set $\vartheta = k$ in our algorithms and assess their performance on cost. Particularly notable are the results in the cost comparison between COCA, and LIB, plotted in Figure 5.9 and 5.10 which confirms a logarithmic reduction using the doubling trick. Note that here, we re-visualize the results for real-world datasets separately for enhanced clarity and additionally provide results on synthetic datasets.

Dataset	LIB	COCA	$\texttt{LIB}_{\texttt{H}}$	COCH
Adult	292.9 ± 20.79	578.9 ± 93.62	9.0 ± 0.0	9.0 ± 0.0
Bank	311.9 ± 33.29	509.4 ± 78.28	9.0 ± 0.0	9.2 ± 0.4
Diabetes	109.5 ± 15.18	131.7 ± 17.92	8.6 ± 0.79	9.1 ± 0.3

Table 5.21: Comparison of $k_{\texttt{actual}}$ when $k_{\texttt{target}}$ is 2 and ϑ is $k_{\texttt{target}}$ (uncapacitated setting) for COCA, COCH against uncapacitated SOTA.



(a) Adult dataset (b) Bank dataset

Figure 5.9: Cost comparison of COCA (unrestricted setting, i.e., when ϑ is $\texttt{k}_{\texttt{target}}$) to LIB. The plots validate the theoretical cost reduction of a logarithmic factor on different datasets: (a) Adult, (b) Bank.

5.7.2 Analysis of Online $\texttt{COCA}_F$ with Fairness as Capacity Constraints:

We now validate the efficacy of the online capacitated algorithm when capacity constraints are provided at a finer level rather than just adding constraints on overall cluster sizes. We

Dataset	LIB	COCA	$\texttt{LIB}_\texttt{H}$	COCH
Adult	474.0 ± 67.56	763.2 ± 103.11	12.8 ± 0.6	13.8 ± 0.87
Bank	522.1 ± 71.14	736.5 ± 98.98	12.1 ± 1.13	13.4 ± 0.79
Diabetes	145.4 ± 22.64	142.3 ± 13.50	12.0 ± 1.09	12.8 ± 0.6

Table 5.22: Comparison of $k_{\texttt{actual}}$ when $k_{\texttt{target}}$ is 3 and ϑ is $k_{\texttt{target}}$ (uncapacitated setting) for COCA, COCH against uncapacitated SOTA.

Dataset	LIB	COCA	$\texttt{LIB}_\texttt{H}$	COCH
Adult	806.7 ± 93.32	1010.4 ± 133.40	20.0 ± 1.34	21.7 ± 1.18
Bank	886.1 ± 115.89	1091.1 ± 133.72	19.7 ± 1.48	21.4 ± 1.2
Diabetes	195.8 ± 26.97	171.1 ± 7.96	17.5 ± 0.80	19.1 ± 1.51

Table 5.23: Comparison of $k_{\texttt{actual}}$ when $k_{\texttt{target}}$ is 5 and ϑ is $k_{\texttt{target}}$ (uncapacitated setting) for COCA, COCH against uncapacitated SOTA.

Dataset	LIB	COCA	$\texttt{LIB}_\texttt{H}$	COCH
Adult	1153.4 ± 159.58	1213.3 ± 131.88	29.3 ± 0.45	29.3 ± 0.45
Bank	1311.6 ± 137.23	1392.5 ± 178.84	26.4 ± 2.01	29.4 ± 0.66
Diabetes	214.4 ± 25.81	164.0 ± 10.23	22.3 ± 0.64	24.9 ± 2.11

Table 5.24: Comparison of $k_{\texttt{actual}}$ when $k_{\texttt{target}}$ is 7 and ϑ is $k_{\texttt{target}}$ (uncapacitated setting) for COCA, COCH against uncapacitated SOTA.

Dataset	LIB	COCA	$\texttt{LIB}_\texttt{H}$	COCH
Adult	1857.4 ± 206.12	1746.1 ± 201.22	36.5 ± 2.57	41.0 ± 0.44
Bank	1969.5 ± 205.36	1748.7 ± 187.29	37.3 ± 3.0	41.3 ± 0.64
Diabetes	246.3 ± 20.34	187.4 ± 15.04	31.1 ± 0.3	33.9 ± 2.70

Table 5.25: Comparison of $k_{\texttt{actual}}$ when $k_{\texttt{target}}$ is 10 and ϑ is $k_{\texttt{target}}$ (uncapacitated setting) for COCA, COCH against uncapacitated SOTA.

Dataset	LIB	COCA	$\texttt{LIB}_\texttt{H}$	COCH
Adult	3143.1 ± 866.86	2208.3 ± 254.00	54.5 ± 5.46	60.9 ± 0.30
Bank	3273.6 ± 504.60	2324.0 ± 249.02	59.2 ± 2.56	63.9 ± 2.91
Diabetes	265.8 ± 4.70	204.3 ± 8.96	50.7 ± 5.86	47.9 ± 2.80

Table 5.26: Comparison of $k_{\texttt{actual}}$ when $k_{\texttt{target}}$ is 15 and ϑ is $k_{\texttt{target}}$ (uncapacitated setting) for COCA, COCH against uncapacitated SOTA.

Dataset	LIB	COCA	$\texttt{LIB}_\texttt{H}$	COCH
Adult	4657.4 ± 1060.78	2566.2 ± 256.22	77.1 ± 5.37	81.0 ± 0.0
Bank	4469.8 ± 544.19	2863.4 ± 312.30	80.3 ± 2.09	82.4 ± 2.10
Diabetes	268.8 ± 1.4	211.1 ± 13.07	71.5 ± 7.81	66.0 ± 6.55

Table 5.27: Comparison of $k_{\texttt{actual}}$ when $k_{\texttt{target}}$ is 20 and ϑ is $k_{\texttt{target}}$ (uncapacitated setting) for COCA, COCH against uncapacitated SOTA.

Dataset	LIB	COCA	LIB_{H}	COCH
Adult	5946.6 ± 1382.66	2857.4 ± 275.42	94.7 ± 7.31	100.3 ± 2.79
Bank	5571.2 ± 701.10	3002.1 ± 247.06	98.6 ± 4.94	102.1 ± 1.04
Diabetes	270.1 ± 1.3	226.1 ± 8.45	124.9 ± 50.44	83.6 ± 7.07

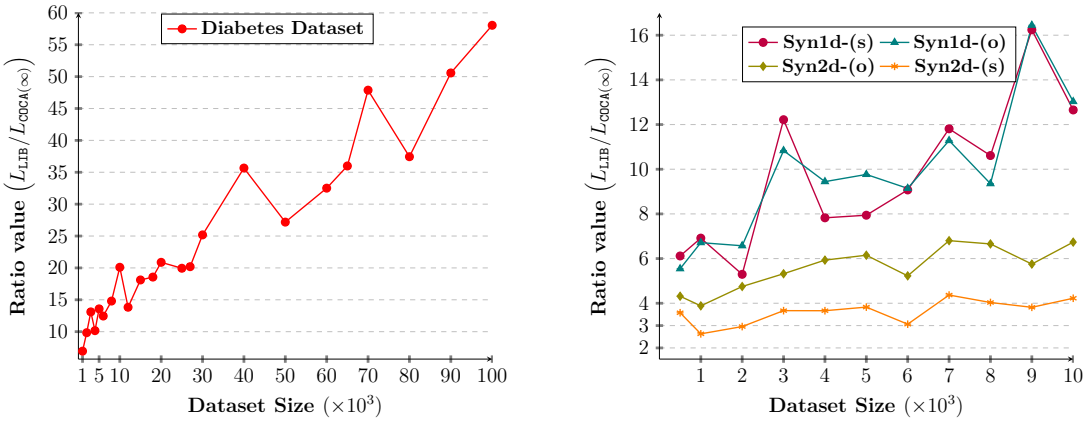
Table 5.28: Comparison of $k_{\texttt{actual}}$ when $k_{\texttt{target}}$ is 25 and ϑ is $k_{\texttt{target}}$ (uncapacitated setting) for COCA, COCH against uncapacitated SOTA.

Dataset	LIB	COCA	LIB_{H}	COCH
Adult	7136.4 ± 1617.56	3075.7 ± 268.01	114.7 ± 8.1	118.7 ± 4.00
Bank	6756.1 ± 834.40	3490.7 ± 320.93	116.2 ± 6.24	122.5 ± 1.5
Diabetes	271.5 ± 1.74	254.3 ± 12.89	198.7 ± 75.14	93.4 ± 2.24

Table 5.29: Comparison of $k_{\texttt{actual}}$ when $k_{\texttt{target}}$ is 30 and ϑ is $k_{\texttt{target}}$ (uncapacitated setting) for COCA, COCH against uncapacitated SOTA.

Dataset	LIB	COCA	LIB_{H}	COCH
Adult	9865.8 ± 1889.41	3592.0 ± 298.70	154.4 ± 8.19	159.6 ± 3.92
Bank	9516.1 ± 1094.63	3907.9 ± 347.39	161.3 ± 2.53	162.0 ± 1.84
Diabetes	274.4 ± 2.53	324.2 ± 6.76	251.8 ± 47.46	116.2 ± 9.38

Table 5.30: Comparison of $k_{\texttt{actual}}$ when $k_{\texttt{target}}$ is 40 and ϑ is $k_{\texttt{target}}$ (uncapacitated setting) for COCA, COCH against uncapacitated SOTA.



(a) Diabetes dataset (b) Synthetic dataset

Figure 5.10: Cost comparison of COCA (unrestricted setting, i.e., when ϑ is $\texttt{k}_{\texttt{target}}$) to LIB. The plots validate the theoretical cost reduction of a logarithmic factor on different datasets: (a) Diabetes, (b) Synthetic.

report the results for COCA_F on the number of centers opened in uniform and uncapacitated settings. We also provide results on other capacity parameter values and report the cost approximation factors as well. The results are summarized below:

Analysis on Number of Centers Opened

–**Under uniform capacities** ($\vartheta = 1$): We compare the centers opened by COCA_F, COCH_F with SOTA. Results for on $k_{\texttt{target}}$ of $2, 3, 5, 7, 10, 15, 20, 25, 30$ and 40 are listed

in Tables 5.31 to 5.40. As can be observed from the tables, the number of centers opened is comparatively lower than COCA and LIB. One possible reason is that, besides having capacity limits for each group value at centers, we also estimate the number of data points and center opening costs separately for each group. These factors together help to have better control over the number of centers while not losing too much on cost. Having separate capacity constraints may appear to restrict the number of data points from a particular group (say ℓ). However, it helps ensure lower costs for data points belonging to other group values. The data point from a group ℓ ends up opening a new center and accommodates future data points. This decision in COCA would have happened once the complete cluster capacity had reached its threshold but would have happened eventually. However, having this decision sooner helps in data points from other group values less probable in the data stream to have a fair chance of getting assigned to their closest centers, which otherwise might have ended up as new centers. Thus, these factors help in controlling the number of centers opened.

For the heuristic method $\texttt{COCH}_F$, the number of centers opened is slightly more than COCH and $\texttt{LIB}_\texttt{H}$. Further, the gap between $k_{\texttt{target}}$, $k_{\texttt{actual}}$ increases as the target increases, considering the rising uncertainty of the arrival order of points. We must note that the rise is still tolerable and results in fewer centers than fully online methods $\texttt{COCA}_F$, COCA.

–**Under relaxed capacities ($\vartheta > 1$)**: We also evaluate the performance of $\texttt{COCA}_F$ as the capacity parameter (ϑ) is relaxed from 1 to 2.5, 5, 7, and $\texttt{k}_{\texttt{target}}$. The results are presented in Tables 5.41 to 5.45 for $\texttt{k}_{\texttt{target}}$ values spanning from 2 to 40. Remarkably, we observe that the number of opened centers remains relatively consistent even as we relax the constraint on ϑ from the most restrictive setting of one. This can be attributed to the fact that the online algorithms, whether capacitated or uncapacitated, open a sufficient number of centers to accommodate all points due to the unknown order of arrival of data points in the online setting. While only at a lower target value of two, there is a slight difference (increase or decrease) in centers opened.

Analysis on Clustering Cost

We further validate our theoretical findings on the cost approximation of fair online capacitated $\texttt{COCA}_F$ and $\texttt{COCH}_F$ to their offline capacitated clustering (Figure 5.11). We observe that theoretically, we achieve constant cost approximation, but experimentally, due to the stochastic nature of the ordering of data points, there is a slight increase in the approximation factor as the target value increases. The main reason behind this is that as the target number of centers increases, the randomness increases, resulting in a higher cost of fairness. We further analyze the cost factor increase when we shift the capacity constraints from the cluster level to applying constraints at each group level. To this, we compare the cost of $\texttt{COCA}_F$ to the cost of COCA in Figure 5.12. The results show that we do not lose much in terms of cost by shifting the focus to applying a more granular level of capacity constraints. Rather, we can see that it helps in opening a lower number of centers while achieving a low cost approximation ratio. For the Diabetes dataset, due to

the existence of local minima [44] and the fact that sometimes the offline algorithm gets stuck in local optima, we observe that the ratio is slightly below one.

Dataset	LIB	COCA	$\texttt{COCA}_F$	$\texttt{LIB}_\texttt{H}$	COCH	$\texttt{COCH}_F$
Adult	292.9±20.79	541.7 ± 79.80	182.0 ± 4.63	9.0±0.0	9.0±0.0	15.75 ± 0.43
Bank	311.9±33.29	544.3 ± 82.69	224.0 ± 7.0	9.0±0.0	9.0±0.0	22.0 ± 0.70
Diabetes	109.5 ± 15.18	143.5 ± 16.55	67.25 ± 1.47	8.6±0.79	9.0±0.0	14.0 ± 1.00

Table 5.31: Comparison against fair $\texttt{COCA}_F$, $\texttt{COCH}_F$ on $k_{\texttt{actual}}$ against various SOTA methods when $k_{\texttt{target}}$ is 2 under uniform capacities (i.e., ϑ is 1).

Dataset	LIB	COCA	$\texttt{COCA}_F$	$\texttt{LIB}_\texttt{H}$	COCH	$\texttt{COCH}_F$
Adult	474.0±67.56	766.4 ± 107.53	340.75 ± 6.83	12.8±0.6	13.8±0.87	24.75 ± 0.83
Bank	522.1±71.14	737.5 ± 98.59	427.0 ± 7.90	12.1±1.13	13.4±0.79	35.75 ± 0.43
Diabetes	145.4±22.64	141.0 ± 13.59	97.25 ± 3.96	12.0±1.09	12.8±0.6	20.75 ± 0.82

Table 5.32: Comparison against fair $\texttt{COCA}_F$, $\texttt{COCH}_F$ on $k_{\texttt{actual}}$ against various SOTA methods when $k_{\texttt{target}}$ is 3 under uniform capacities (i.e., ϑ is 1).

Dataset	LIB	COCA	$\texttt{COCA}_F$	$\texttt{LIB}_\texttt{H}$	COCH	$\texttt{COCH}_F$
Adult	806.7±93.32	1012.1 ± 133.66	528.25 ± 12.19	20.0±1.34	21.7±1.18	40.25 ± 0.43
Bank	886.1±115.89	1095.7 ± 133.58	768.5 ± 27.59	19.7±1.48	21.4±1.2	57.5 ± 1.50
Diabetes	195.8±26.97	170.5 ± 12.75	125.50 ± 7.63	17.5±0.80	19.1±1.51	31.0 ± 0.70

Table 5.33: Comparison against fair $\texttt{COCA}_F$, $\texttt{COCH}_F$ on $k_{\texttt{actual}}$ against various SOTA methods when $k_{\texttt{target}}$ is 5 under uniform capacities (i.e., ϑ is 1).

Dataset	LIB	COCA	$\texttt{COCA}_F$	$\texttt{LIB}_\texttt{H}$	COCH	$\texttt{COCH}_F$
Adult	1153.4±159.58	1220.5 ± 126.19	748.75 ± 10.96	26.3±1.73	29.3±0.45	56.25 ± 0.43
Bank	1311.6±137.23	1388.1 ± 180.04	1010.25 ± 48.43	26.4±2.01	29.4±0.66	87.75 ± 4.60
Diabetes	214.4±25.81	163.8 ± 8.07	135.0 ± 4.84	22.3±0.64	24.9±2.11	42.25 ± 0.43

Table 5.34: Comparison against fair $\texttt{COCA}_F$, $\texttt{COCH}_F$ on $k_{\texttt{actual}}$ against various SOTA methods when $k_{\texttt{target}}$ is 7 under uniform capacities (i.e., ϑ is 1).

Dataset	LIB	COCA	$\texttt{COCA}_F$	$\texttt{LIB}_\texttt{H}$	COCH	$\texttt{COCH}_F$
Adult	1857.4±206.12	1735.8 ± 202.95	1120.0 ± 28.23	36.5±2.57	41.0±0.44	78.25 ± 3.03
Bank	1969.5±205.36	1751.4 ± 191.69	1356.75 ± 39.78	37.3±3.00	41.3±0.64	119.50 ± 4.71
Diabetes	246.3±20.34	189.0 ± 16.05	163.75 ± 15.20	31.1±0.3	33.9±2.7	234.0 ± 174.01

Table 5.35: Comparison against fair $\texttt{COCA}_F$, $\texttt{COCH}_F$ on $k_{\texttt{actual}}$ against various SOTA methods when $k_{\texttt{target}}$ is 10 under uniform capacities (i.e., ϑ is 1).

Dataset	LIB	COCA	$\texttt{COCA}_F$	$\texttt{LIB}_\texttt{H}$	COCH	$\texttt{COCH}_F$
Adult	3143.1±866.86	2205.6 ± 251.65	1599.0 ± 56.10	54.5±5.46	60.9±0.3	121.75 ± 3.63
Bank	3273.6±504.60	2325.3 ± 246.43	2085.75 ± 27.38	59.2±2.56	63.9±2.91	187 ± 14.61
Diabetes	265.8±4.70	206.0 ± 7.94	188.0 ± 11.33	50.7±5.86	47.9±2.80	792.50 ± 177.32

Table 5.36: Comparison against fair $\texttt{COCA}_F$, $\texttt{COCH}_F$ on $k_{\texttt{actual}}$ against SOTA methods when $k_{\texttt{target}}$ is 15 under uniform capacities (i.e., ϑ is 1).

Dataset	LIB	COCA	$\texttt{COCA}_F$	$\texttt{LIB}_\texttt{H}$	COCH	$\texttt{COCH}_F$
Adult	4657.4±1060.78	2579.4 ± 258.59	1907.75 ± 40.28	77.1±5.37	81.0±0.0	163.0 ± 2.54
Bank	4469.8±544.19	2867.3 ± 309.31	2509.25 ± 72.76	80.3±2.09	82.4±2.11	244.75 ± 17.10
Diabetes	268.8±1.4	211.3 ± 15.58	193.0 ± 13.49	71.5±7.81	66.4.1±6.74	648.0 ± 388.38

Table 5.37: Comparison against fair $\texttt{COCA}_F$, $\texttt{COCH}_F$ on $k_{\texttt{actual}}$ against SOTA methods when $k_{\texttt{target}}$ is 20 under uniform capacities (i.e., ϑ is 1).

Dataset	LIB	COCA	$\texttt{COCA}_F$	$\texttt{LIB}_\texttt{H}$	COCH	$\texttt{COCH}_F$
Adult	5946.6±1382.66	2858.7 ± 284.03	2230.25 ± 57.97	94.7±7.31	100.3±2.79	203.50 ± 3.20
Bank	5571.2±701.10	3007.6 ± 256.45	2739.75 ± 27.34	98.6±4.94	102.1±1.04	302.25 ± 4.26
Diabetes	270.1±1.3	224.4 ± 9.44	950.25 ± 335.21	124.9±50.44	83.6±6.96	1354.75 ± 350.09

Table 5.38: Comparison against fair $\texttt{COCA}_F$, $\texttt{COCH}_F$ on $k_{\texttt{actual}}$ against SOTA methods when $k_{\texttt{target}}$ is 25 under uniform capacities (i.e., ϑ is 1).

Dataset	LIB	COCA	$\texttt{COCA}_F$	$\texttt{LIB}_\texttt{H}$	COCH	$\texttt{COCH}_F$
Adult	7136.4±1617.56	3071.1 ± 251.07	2459.0 ± 22.98	114.7±8.1	118.7±4.00	240.0 ± 6.48
Bank	6756.1±834.40	3485.0 ± 335.15	3101.50 ± 69.64	116.2±6.24	122.9 ± 1.58	351.75 ± 14.95
Diabetes	271.50±1.74	252.7 ± 12.46	360.20 ± 92.88	198.7±75.14	93.6±2.42	3910.25 ± 125.59

Table 5.39: Comparison against fair $\texttt{COCA}_F$, $\texttt{COCH}_F$ on $k_{\texttt{actual}}$ against various SOTA methods when $k_{\texttt{target}}$ is 30 under uniform capacities (i.e., ϑ is 1).

Dataset	LIB	COCA	$\texttt{COCA}_F$	$\texttt{LIB}_\texttt{H}$	COCH	$\texttt{COCH}_F$
Adult	9865.8±1889.41	3594.8 ± 294.33	2913.75 ± 47.09	154.4±8.19	159.6±3.93	388.0 ± 8.15
Bank	9516.1±1094.63	3744.7 ± 576.77	3578.0 ± 141.30	161.3±2.53	162.1±1.92	463.75 ± 9.09
Diabetes	274.4 ± 2.53	323.3 ± 7.87	320.0 ± 84.91	251.8±47.46	117.0±9.04	116.0 ± 14.93

Table 5.40: Comparison against fair $\texttt{COCA}_F$, $\texttt{COCH}_F$ on $k_{\texttt{actual}}$ against SOTA methods when $k_{\texttt{target}}$ is 40 under uniform capacities (i.e., ϑ is 1).

Dataset	$\vartheta = 1$	$\vartheta = 2.5$	$\vartheta = 5$	$\vartheta = 7$	$\vartheta = \texttt{k}_{\texttt{target}}$
Adult	182.0 ± 4.63	194.25 ± 13.75	160.50 ± 12.77	221.25 ± 14.51	184.25 ± 13.31
Bank	224.0 ± 7.0	249.0 ± 16.65	251.75 ± 7.36	260.25 ± 12.23	239.25 ± 12.77
Diabetes	67.25 ± 1.47	71.50 ± 2.69	73.25 ± 3.69	72.25 ± 2.86	84.75 ± 3.89

Table 5.41: $k_{\texttt{actual}}$ on $\texttt{COCA}_F$ methods when $k_{\texttt{target}}$ is 2 under varying capacity parameter.

Dataset	$\vartheta = 1$	$\vartheta = 2.5$	$\vartheta = 5$	$\vartheta = 7$	$\vartheta = \mathtt{k_{target}}$
Adult	340.75 ± 6.83	340.75 ± 6.83	340.75 ± 6.83	340.75 ± 6.83	24.75 ± 0.83
Bank	427.0 ± 7.90	427.0 ± 7.90	427.0 ± 7.90	427.0 ± 7.90	427.0 ± 7.90
Diabetes	97.25 ± 3.96	97.25 ± 3.96	97.25 ± 3.96	97.25 ± 3.96	97.25 ± 3.96

Table 5.42: $k_{\mathtt{actual}}$ on $\mathtt{COCA}_F$ methods when $k_{\mathtt{target}}$ is 3 under varying capacity parameter.

Dataset	$\vartheta = 1$	$\vartheta = 2.5$	$\vartheta = 5$	$\vartheta = 7$	$\vartheta = \mathtt{k_{target}}$
Adult	1120.0 ± 28.23	1120.0 ± 28.23	1120.0 ± 28.23	1120.0 ± 28.23	1120.0 ± 28.23
Bank	1356.75 ± 39.78	1356.75 ± 39.78	1356.75 ± 39.78	1356.75 ± 39.78	1356.75 ± 39.78
Diabetes	163.75 ± 15.20	163.75 ± 15.20	163.75 ± 15.20	163.75 ± 15.20	163.75 ± 15.20

Table 5.43: $k_{\mathtt{actual}}$ on $\mathtt{COCA}_F$ methods when $k_{\mathtt{target}}$ is 10 under varying capacity parameter.

Dataset	$\vartheta = 1$	$\vartheta = 2.5$	$\vartheta = 5$	$\vartheta = 7$	$\vartheta = \mathtt{k_{target}}$
Adult	2230.25 ± 57.94	2230.25 ± 57.94	2230.25 ± 57.94	2230.25 ± 57.94	2230.25 ± 57.94
Bank	2739.75 ± 27.34	2739.75 ± 27.34	2739.75 ± 27.34	2739.75 ± 27.34	2739.75 ± 27.34
Diabetes	201.75 ± 3.26	201.75 ± 3.26	201.75 ± 3.26	201.75 ± 3.26	201.75 ± 3.26

Table 5.44: $k_{\mathtt{actual}}$ on $\mathtt{COCA}_F$ methods when $k_{\mathtt{target}}$ is 25 under varying capacity parameter.

Dataset	$\vartheta = 1$	$\vartheta = 2.5$	$\vartheta = 5$	$\vartheta = 7$	$\vartheta = \mathtt{k_{target}}$
Adult	2913.75 ± 47.09	2913.75 ± 47.09	2913.75 ± 47.09	2913.75 ± 47.09	2913.75 ± 47.09
Bank	3578.0 ± 141.30	3578.0 ± 141.30	3578.0 ± 141.30	3578.0 ± 141.30	3578.0 ± 141.30
Diabetes	313.0 ± 8.51	313.0 ± 8.51	313.0 ± 8.51	313.0 ± 8.51	313.0 ± 8.51

Table 5.45: $k_{\mathtt{actual}}$ on $\mathtt{COCA}_F$ methods when $k_{\mathtt{target}}$ is 40 under varying capacity parameter.

5.8 Conclusion

This work extends the probabilistic algorithm available in uncapacitated to capacitated online clustering. Our algorithm (COCA) is the first online algorithm to tackle capacity constraints in h-dimensional space for k-means or k-median. We introduce two novel changes to existing online uncapacitated clustering: First, we determine the initial number of centers to be opened by the algorithm to get a better representation, and second, we employ a doubling trick to estimate the total number of data points. These changes result in fewer centers opening while achieving constant cost approximation to the optimal clustering problem. We further extend COCA to accommodate for group fairness constraints and propose $\mathtt{COCA}_F$. The algorithm undergoes experimental analysis on the number of centers opened and cost. The results provide experimental validation of $\mathtt{COCA}_F$'s cost approximation guarantees. An immediate future direction involves extending the work in the presence of noisy data. Another interesting problem is the extension to group fair assignments [44] or centers [49]. Also, since capacity constraints and group fairness in online streaming can result in different assignments compared to offline counterparts, focusing on minimizing such reassignments is interesting.

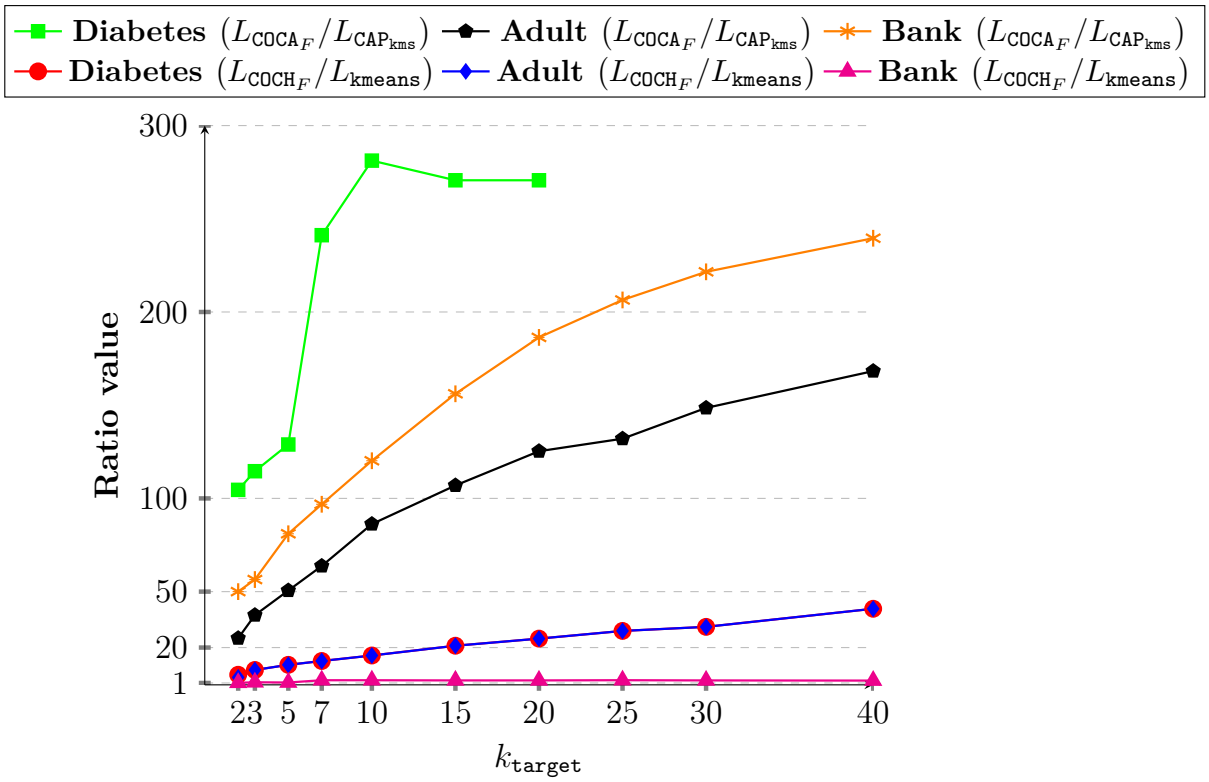


Figure 5.11: Cost approximation of $\texttt{COCA}_F$ to offline capacitated k-means clustering ($\texttt{CAP}_{\texttt{kms}}$). Additionally, provide cost approximation of fair online clustering heuristic $\texttt{COCH}_F$ to uncapacitated offline k-means.

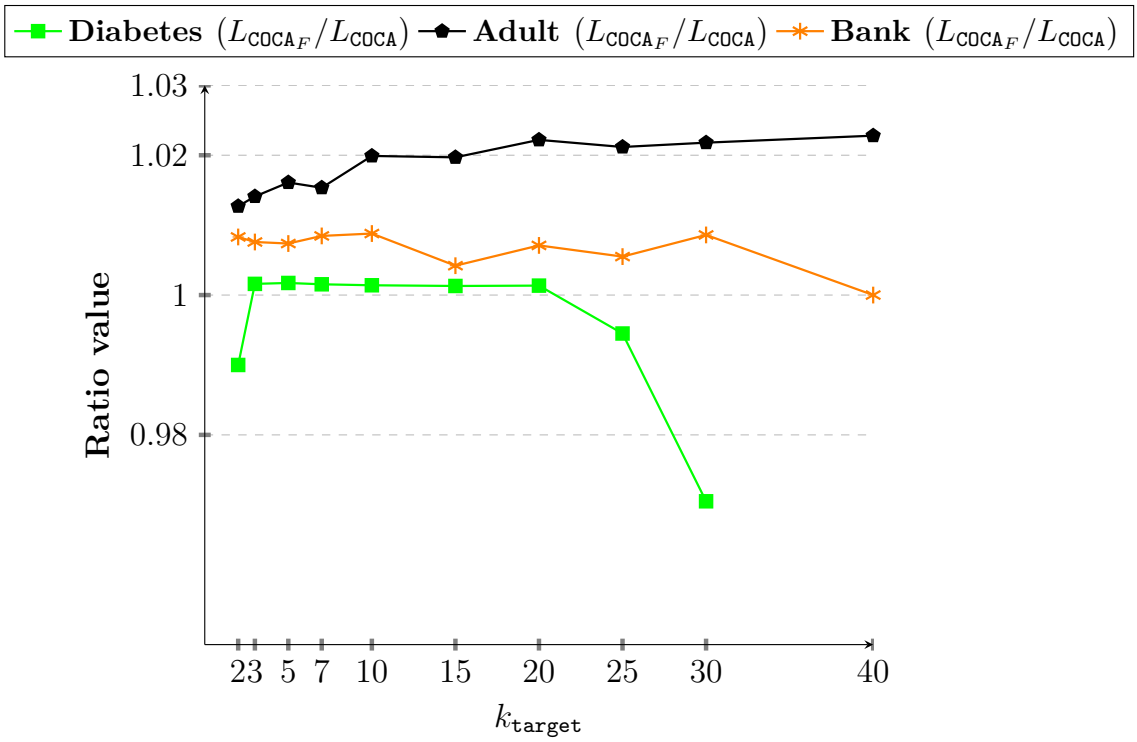


Figure 5.12: Cost approximation of fair $\texttt{COCA}_F$ to online unfair ($\texttt{COCA}$).

Chapter 6

Algorithms for Efficient and Fair Federated Data Clustering

Abstract

The rapid growth of data has catalyzed the performance of machine learning (ML) algorithms. However, with the rising concerns about data privacy, traditional ML faces two major challenges - the centralization of the data for training and the non-availability of labels in the data. To overcome these issues, initial attempts have been to solve distributed privacy-preserving unsupervised 'federated data clustering'. The goal is to partition the data available on clients into k partitions (called clusters) without the actual exchange of the data points. Most of the existing algorithms are highly dependent on data distribution patterns across clients or are computationally expensive. To this, we are first to propose a multi-shot approach called `MFC`*. The* `MFC`*'s performance is independent of the underlying client data distribution. We also theoretically show that cluster centers obtained using* `MFC` *are not too far from the optimal centers. Additionally, if the number of clients is at least* $O(k^2 \log k)$*, then* `MFC` *can achieve stricter privacy on the shared local information while having similar performance guarantees. Furthermore, due to skewness in data distribution, clients may suffer high clustering costs and may leave the system. In order to prevent this, we are first to introduce the idea of personalization in federated clustering and propose an improvisation of* `MFC` *called* `p-FClus`*. It ensures a uniform cost distribution across clients in a single round of communication between server and clients. Both* `p-FClus` *and* `MFC` *undergo extensive experimentation on various synthetic and real-world datasets. The results showcase their efficacy on cost, data-independent nature and applicability to any finite norm value while* `p-FClus` *additionally achieves lower cost variance across clients.*

6.1 Introduction

In the age of rapid technological advancement, the proliferation of devices (such as smartphones, tablets, Internet of Things sensors, and edge devices) has become an inherent aspect of our daily lives [265]. This surge in technological devices has resulted in an exponential growth of data [266], which has significantly impacted many domains,

A preliminary part of this chapter has appeared in the European Conference on Artificial Intelligence (ECAI) 2023 [108]. A detailed version of this chapter is under review.

especially machine learning (ML). The primary reason is that many ML algorithms perform better with more data [267]. Furthermore, the abundance of data facilitates training more complex, robust and generalized models, thus underscoring the notion that 'data is the new oil'. However, traditional machine learning methods necessitate centralized data collection for model training, typically on a server. But as, most of the existing generated data resides outside the servers and data centers (i.e., on edge devices) and is private. Thus, concerns about data privacy and protection during collection at centralized repositories are rising among alliances, governments, and researchers about data privacy and protection during collection at server [268].

In response to this challenge, Federated Learning (FL) is emerging as a promising solution for collaborative ML model training without centralized data collection. Pioneered by Google Inc. in 2016, FL revolutionizes the traditional model training process into a novel distributed paradigm [269]. In this approach, the server shares with each participating client (or edge device) an initial model usually trained on publicly available data points. As with the passage of time, more private data points are generated at each client; they are then tasked with training their respective models (referred to as local models) on newly generated private data. Subsequently, clients share model parameter updates instead of sharing the complete updated models or local data with the server. After gathering updates from all clients[1], the server aggregates these and computes a global model update. This global parameter update is then shared back with clients for future use. While the existing literature encompasses various aggregation policies, the simplest involves computing the weighted average of model parameters (called FedAvg [269]) based on the local dataset size of each client. As communication involves solely sharing model updates rather than actual data points, the risk of recovering the original data points from these updates is minimized. Although a few works suggest that FedAvg is still prone to data recovery attacks, the literature offers numerous more robust, privacy-enhanced methods [270, 271, 272, 273, 274]. Consequently, one can assert that FL is emerging as a burgeoning paradigm for harnessing the hidden potentials inherent within the growing data landscape.

Within the field of FL, two prominent frameworks, namely Horizontal Federated Learning (HFL) and Vertical Federated Learning (VFL), have gathered significant attention over the past few years [275]. These frameworks differ in how data points are split among participating clients. In the HFL framework, data instances on client devices share the same set of attributes and labels [275]. On the contrary, in VFL, clients exhibit a larger overlap in data instances but a less similar set of attributes (feature set and labels) [275]. Although the literature extensively studies supervised federated learning, where the feature set and labels are available in data instances [276, 277, 267]. However, in practical, real-life scenarios, one often faces challenges, as data available on the clients may lack labels [21]. A few possible reasons for this include a lack of motivation, incentives, or expertise to label their data points. For instance, in scenarios like the clustering of social media posts

[1] A few literature work opts for a subset of clients to improve throughput and efficiency

for sentiment analysis, clients might be reluctant to invest efforts to label their posts as happy or sad [278, 279]. Furthermore, even if clients are willing to label their data points, they might not fully label all the local data points, potentially due to a lack of expertise in labelling. This challenge is also evident in emerging smart healthcare, such as wearable fitness bands, where clients may lack accurate knowledge to label themselves as medically fit or unhealthy based on readings from different sensors (such as heart rate, blood oxygen level, and sugar level) [280]. This raises the key question: How can one find a global model in a federated environment when confronted with unlabelled data?
To answer this, a few initial attempts have been made to handle unlabelled data in various domains such as Speech Enhancement [281], Intrusion Detection [282], Healthcare [283], Driving Style Recognition [284], Synthetic Data Generation [114], Recommender Systems [285, 286] and Autonomous Driving [287]. The solution approaches used in these directions can be categorized into two types:

1. The methods that assume the availability of a limited amount of labelled data at clients, i.e., Federated Semi-Supervised Learning (FedSSL). These methods involve using (a) pre-trained models to annotate clients' local data [288, 289], (b) providing pseudo-labels [290, 291], and (c) fine-tuning with labelled data [292, 287, 293].

2. The more challenging and intriguing problem of tackling situations where no labelled data is available i.e., Federated Unsupervised Learning (FedUL).

The focus of this chapter will be to better understand the later setting. Primarily, we will investigate clustering in the FedUL setting. Federated clustering mainly involves dividing the data points available to clients into k partitions (called clusters). Federated clustering finds applications in numerous domains [294, 43, 223, 295, 296], one of which is as follows - Consider a situation where multiple banks want to cluster their users' transaction data to differentiate legitimate transactions from fraudulent ones. In such a situation, certain banks may even have limited samples of fraudulent transactions, and due to security and privacy concerns, banks are prohibited from sharing their data with each other. Therefore, having a federated data clustering model can enable banks to reap the benefits of collective learning [26, 297, 298]. Note that the existing solutions in supervised FL cannot be directly mapped to FedUL clustering as it primarily involves the following challenges:

(a) Each client may not contain data points from all k partitions.

(b) As in supervised FL, there does not exist a chronological ordering[2] (or synchronization) on cluster centers, so mapping centers for an averaging function is non-trivial and can lead to bad initialization [110].

Researchers have undertaken various studies in existing literature to overcome the above challenges. An initial attempt is in Dennis et al. [41] (`k-FED`), which extends the centralized method from Awasthi and Sheffet [109] to the federated setting in two steps:

[2]Unlike supervised learning, where fixed ordering or numbering of weights exists across clients, no such numbering exists for cluster centers in unsupervised clustering.

(1) Apply the centralized method locally on clients. (2) Share the local information about centers to the server before reapplying the clustering therein. While `k-FED` demonstrates single-round communication efficiency, it faces difficulties in certain scenarios as the method is highly dependent on the nature of data distribution across clients (as evident from our experimental study, Section 6.7 as well). Some other directions explore synthetic data generation or utilize encryption techniques, albeit at the cost of increased computational expenses [115]. To this, we propose (called `MFC`) an enhanced `k-FED` that reduces the one-shot communication load on clients and leverages the advantage of multiple communication rounds with minimal information exchange. The approach also addresses the challenge of data distribution dependence, as seen in `k-FED`.

It is also important to note that a more common limitation across all existing methods is that the best local centers may be distant from the global centers, leading to high-cost deviations across clients. The main reason for this high cost is that the global centers may not generalize well, potentially due to non-identically and independently (non-iid) distributed data points across clients, sometimes even resulting in highly skewed distributions. This phenomenon can lead to an inductive effect and the potential reluctance of clients to contribute to the federated system. Thus, there is a pressing need to extend `MFC` further to ensure that the centers provided to clients are not too far from local ones. This, in turn, ensures lower cost deviation across clients and long-term commitment to the system. For example, in a banking scenario, banks may need to adjust a global clustering model to suit local factors such as user intelligence, fraudulent behavior, income levels, and fraud amounts. This problem can be formulated as developing personalized clustering (close to local) models.

We are the first to address this open direction in federated data clustering and propose another method, which we call `p-FClus` (**p**ersonalized-**F**ederated **Clus**tering). The algorithm handles all the prior challenges as those handled by `MFC`. Broadly, the algorithm primarily involves three steps: firstly, finding the initial local cluster models, which are then used to build a collaborative global model. The last step involves specialized unsupervised fine-tuning using center mapping and point-wise gradient updates on clients' local data. This helps in achieving individual personalized models. To sum up, overall, our **contributions** in this chapter are as follows-

- We propose `MFC` that unlike `k-FED` does not rely on the data distribution across clients and under well separability assumptions (similar to [109, 41]), we have theoretical bounds on the gap between local and global centers obtained using `MFC`.
- Motivated by the performance of `MFC`, we propose another method `p-FClus` that exhibits lower or comparable cost deviation across clients, leading to a fairer and more personalized solution. The method is the first attempt to provide personalization in federated data clustering.
- We experimentally validate the efficacy of both approaches on a variety of datasets. Results show that `p-FClus` achieves a lower clustering objective cost in a single round

of communication and, secondly, is independent of the nature of data distribution (or division) among clients. Thus, `p-FClus` captures the benefits of both state-of-the-art (`SOTA`) `k-FED` (single shot) and `MFC` (data distribution independence), resulting in an efficient algorithm.

Organization: The rest of the chapter is organized as follows: Section 6.2 revisits the current literature on FedUL with a primary focus on federated data clustering. Section 6.3 provides an account of the different notations and definitions that will help readers better understand the chapter. These notations and definitions will be used throughout the chapter to familiarize readers with the proposed algorithms, `MFC` and `p-FClus` in Section 6.4 and 6.6, respectively. Section 6.5 discusses the theoretical guarantees of `MFC`. Next, Section 6.7 discusses the experimental setup and the datasets used for validating the efficacy of `p-FClus`, `MFC` against state-of-the-art (`SOTA`) algorithms on different metrics. Finally, Section 6.8 concludes the work with possible directions to work upon.

6.2 Related Work

ML encompasses a wide range of learning paradigms such as supervised, semi-supervised, and unsupervised learning [299, 300, 301]. While abundant literature is available in supervised FL [277, 276, 302, 303, 304, 305, 306, 307, 308, 309, 310, 311], the works in FedUL are in its nascent stages. With the increasing rise of unlabelled data points [312], FedUL approaches have broadened to include federated clustering. Dennis et al. [41] makes an initial effort to solve federated clustering and proposes an algorithm called `k-FED`. The algorithm builds upon the centralized Awasthi and Sheffet [109] aka (Awasthi). The method assumes that the centers are well separated and clusters follow Gaussian distribution properties. Despite `k-FED`'s single-round communication efficiency, our experiments show that the cost of clustering with `k-FED` can be considerably high for some sets of clients. This can potentially lead to a lack of motivation for continued participation. Yang et al. [110] propose a slightly enhanced greedy centroid-based initialization for `k-FED` which surpasses centralized k-means in specific scenarios.
Some works in this direction approach the problem by framing it as a generative data synthesis problem [111, 40, 112, 113, 114]. The broader picture involves training multiple Generative Adversarial Networks (GANs) locally at clients and utilizing their parameters to construct a global GAN model. This global GAN model is then employed to generate synthetic data and further identify k distinct cluster centers at the server. These centers are subsequently communicated back to clients to partition their local data points. Note that these approaches differ from ours, as we work directly with original data points and aim to identify the best possible centers. Li et al. [115] also pursues a parallel approach to develop privacy-preserving distributed clustering by incorporating concepts from cryptography. The proposed method initially computes local center updates and then shares the encrypted information of centers using Lagrange encoding back to the server. Subsequently, the server then aggregates all secret distance codes from the clients.

The too and fro communication happens till the model converges or a user-defined upper threshold on rounds is achieved. While the algorithm harnesses the advantages of encryption-decryption to safeguard data privacy, such techniques entail substantial computation overhead and communication costs, thereby hindering the scalability of the approach. Similarly, Leeuw [116] employs federated data clustering within the blockchain's committee-based consensus protocol. However, the additional overheads counterbalance the performance improvements.

Note that no existing work in federated data clustering has specifically focused on addressing the challenge of cost distribution spread across clients and fostering a more equitable clustering as a primary goal by leveraging the principles of personalization from a supervised setting [271]. Furthermore, extending these principles to an unsupervised setting is non-trivial due to the lack of chronological ordering[3] on centers and the varying data division across clients. We will address this direction of achieving fairness in the present chapter as well.

Additional Research Works: Recent studies have investigated federated data clustering in the soft (or better called fuzzy) clustering paradigm, wherein a data point can have membership to more than a single cluster [313, 314, 315, 316]. These algorithms are highly dependent on rounding methods used for deployment in real-world applications. Therefore, these works differ slightly from ours as we focus on hard assignments. Furthermore, a few works [317, 318, 319] extend a variant of DBSCAN for the federated setting, where clustering relies on characteristics such as the density of data points in the space. Typically, these methods struggle with high-dimensional data and lack control over the number of clusters. In contrast, the primary focus of the current work is to extend k-centroid clustering to the federated setting. A specific enhancement involves clustering image datasets in a federated environment by leveraging the additional advantages of incorporating the latent representation of these images using encoders. These approaches increase a major portion of the computational load onto the client devices, which are now tasked with training encoders (utilizing backbone networks like ResNet18 [320]). Furthermore, heavy communication bandwidth is required between clients and servers as it involves sharing information about centres and encoder parameters [321, 322]. Similarly, works in [323, 324] investigate federated data clustering for Gaussian Mixture Models (GMM) and Expectation Maximation (EM) algorithms, respectively. It is important to note that the need for having personalized models for all clients in probabilistic and peer-to-peer networks is studied in a few works [324, 325]. Further, Zhang and Xu [325] proposes a cloud-based decentralized, personalized federated averaging framework. These works also motivate the need of personalized methods in distributed federated clustering. It is essential to note that our focus lies in clustering data in a federated setting, which differs from federated client clustering. The latter entails smartly selecting a subset of clients for model updates [326, 327, 328, 329, 330, 331, 332].

[3]Unlike supervised learning, where fixed ordering or numbering of weights exists across clients, no such numbering exists for centers in unsupervised clustering for direct averaging.

6.3 Preliminaries

Let $X \subseteq \mathbb{R}^h$ be a set of data points that are distributed among Z clients in a federated setting and $X^{(z)}$ be the data points on any client $z \in [Z]$. Each data point in $X^{(z)}$ is a h-dimensional real-valued feature vector. We assume that these data points are embedded in a metric space having distance metric $d : X \times X \rightarrow \mathbb{R}^+ \cup \{0\}$ that measures dissimilarity between data points using any p-norm represented as $||\cdot||_p$. Note that the data points in X belong to $[k]$ different true distributions, and the goal of any clustering algorithm is to partition the data points spread across clients into a set of disjoint sets (called clusters) represented by the set of global centers denoted by set $C^g = \{c_1^g, c_2^g, \ldots, c_k^g\}$. The computation of finding these global centers involves initially computing the best local centers that partition the local data $X^{(z)}$ (for any $z \in [Z]$) into k disjoint sets represented by $C^{(z)} = \{c_1^{(z)}, c_2^{(z)}, \ldots, c_k^{(z)}\}$. We denote the local assignment function at each client over any center set, say $C^{(z)}$ by $\phi^{(z)}_{C^{(z)}} : X^{(z)} \rightarrow C^{(z)}$. Note that the data points on any client z may not be sampled from all $[k]$ true distributions, and this idea is captured using the notion of heterogeneity in federated settings. Formally, it is defined as follows:

Definition 6.0 (Heterogeneity)

Given k, the heterogeneity level ($\mathtt{H}$) determines the maximum number of distributions the data points $X^{(z)}$ on a client $z \in [Z]$ belongs to, i.e., $\mathtt{H} \leq k$.

In practice, determining the exact level of heterogeneity ($\mathtt{H}$) on a client is often not feasible. Consequently, a common approach in federated data clustering literature is to compute k ($\geq \mathtt{H}$) partitions on each client [41, 108]. These partitions are not arbitrary selections but are the one that minimizes the following objective cost using final converged global centers:

Definition 6.1 (Objective Cost)

Given k, $\cup_{z\in[Z]} X^{(z)}$, and distance metric $d : X \times X \rightarrow \mathbb{R}^+ \cup \{0\}$ with norm value p the local objective cost $L_p^{(z)}$ of client z of (k, p)-clustering in a federated setting with a set of centers C is computed as follows:

$$L_p^{(z)}(C) = \left(\sum_{x_i \in X^{(z)}} \left(d(x_i, \phi_C^{(z)}(x_i)) \right)^p \right)^{1/p} \tag{6.1}$$

In a federated setup, comparing methods based on the **mean objective cost per data point** is often more realistic than the total objective cost at a client. The primary reason is that dataset sizes across clients can differ significantly in federated settings. Thus, evaluating the per-point cost incurred by clients makes more sense. Mathematically, this can be formulated as follows:

$$\boldsymbol{\mu}^{(z)}(C^g) = \frac{L_p^{(z)}(C^g)}{|X^{(z)}|} \tag{6.2}$$

where $\boldsymbol{\mu}^{(z)}(C^g)$ is the mean cost per data point on any client z Also, further note that in a federated setting, the objective cost suffered by any client z can significantly differ from that of other clients, owing to the fact that data points from $\mathtt{H} \leq k$ distributions can be distributed (or generated) in a highly skewed manner among (or at) clients. Therefore, if the global centers deviate too much from the best local centers, clients might feel reluctant to contribute to the federated environment. Thus, the aim is to not solely focus on minimizing objective cost (or per point cost) but rather to find a (k, p)-clustering in the federated setting that is fair for all clients, i.e., one which achieves near uniform cost across all clients. We formally define such a clustering as follows:

Definition 6.2 (Fair Federated Clustering)

Given that data points are sampled from k true clusters and are distributed over Z clients. Then, for any two set of federated global centers C_1^g and C_2^g, we say that C_1^g is more fair than C_2^g if the **cost deviation per data point** (σ) is lower for C_1^g than C_2^g. Here $\boldsymbol{\sigma}$ over centers C_i^g for $i \in \{1, 2\}$ is given as follows:

$$\boldsymbol{\sigma}(C_i^g) = \sqrt{\frac{\sum_{z \in [Z]} \left(\boldsymbol{\mu}^{(z)}(C_i^g) - \boldsymbol{\mu}(C_i^g)\right)^2}{Z}} \tag{6.3}$$

Note that here $\boldsymbol{\mu}(C^g)$ is the mean value of $\boldsymbol{\mu}^{(z)}(C^g)$ across all clients.

The notion captures the idea analogous to individual fairness and demands that the federated clustering model should treat all clients similarly. We now present our proposed algorithms after covering the preliminary notations and definitions. We first discuss our `MFC` algorithm that leverages the benefits of multiple rounds of communication between server and clients to find a better clustering independent of data distribution across clients. We later extend the `MFC` method to achieve a fairer algorithm called `p-FClus`.

6.4 Multishot Federated Clustering (`MFC`)

`MFC` first runs Algorithm 11 on local data of each client $z \in [Z]$ to obtain a set of k initial local centers denoted by $C^{(z)}$. Each client then transmits its respective set to the server, and the server applies the Lloyd k-means algorithm [333] on collected set S. The obtained global centers C^g are sent back to clients for further update.

After the initial handshake between the client and server, they engage in multiple rounds of communication as follows- Clients use the global centers to update their local assignment functions $\phi^{(z)}$ by re-assigning each data point $x_i \in X^{(z)}$ to the nearest center. The local cluster centers are updated by taking the mean of data points assigned to each center. The client sends the local cluster center $C^{(z)}_{\max}$ back with the maximum clustering cost on local data. Conversely, when the server receives the maximum cost centers, it recalculates the global centers C^g by using Lloyd's k-means on the previous global centers and $C^{(z)}_{\max}$ from all clients. After finding the updated global centers, they are returned to the clients. This

iterative process is repeated for a few rounds until convergence is reached. The complete algorithm for `MFC` is described in Algorithm 12. We now provide theoretical bounds on obtained global cluster centers.

Algorithm 11: Awasthi and Sheffet [109] (centralized)

Input: set of data points X, number of clusters k, maximum iterations T
Output: cluster centers $C = \{c_1, c_2, \ldots, c_k\}$ and assignment function ϕ

Project X onto the subspace spanned by the top k singular vectors. Run any standard approximation algorithm [333] for the k-means problem on the projected matrix $\hat{X}$, and obtain k centers $\{c_1, c_2, \ldots, c_k\}$.
for $i \leftarrow 1$ **to** k **do**
  Set $S_i \leftarrow \left\{\hat{x} : \|\hat{x} - c_i\|_2 \leq \frac{1}{3}\|\hat{x} - c_j\|_2, \forall j \in [k]\right\}$
  Update $c_i \leftarrow \frac{1}{|S_i|}\sum_{\hat{x}\in S_i} \hat{x}$
  Set $\phi(\hat{x}) \leftarrow c_i, \forall \hat{x} \in S_i$
end
$itr = 0$
while *until convergence and* $itr \leq T$ **do**
  for $i \leftarrow 1$ **to** k **do**
    Set $U_i \leftarrow \{\hat{x}_i : \|\hat{x}_i - c_i\|_2 \leq \|\hat{x}_i - c_j\|_2, \forall j \in [k]\}$
    Update $c_i \leftarrow \frac{1}{|U_i|}\sum_{\hat{x}\in U_i} \hat{x}$
    Set $\phi(\hat{x}) \leftarrow c_i, \forall \hat{x} \in U_i$
    $itr \leftarrow itr + 1$
  end
end
return ϕ, $C = \{c_1, c_2, \ldots, c_k\}$

6.5 Theoretical Results for `MFC`

We prove the correctness of our algorithm by showing that the centers obtained from `MFC` are *close* to the oracle clustering. We do so by providing a series of lemmas to prove our main Theorem 6.6 that bounds the distance between these centers.

6.5.1 Assumptions

Before presenting the theoretical proofs, we carefully outline the assumptions on which our analysis is based. Let $\mathcal{C}^* = \{\mathcal{C}_1^*, \mathcal{C}_2^*, \ldots, \mathcal{C}_k^*\}$ denote the optimal clustering of datapoints X with centers $\{c_1^*, c_2^*, \ldots, c_k^*\}$. Let n_i denote the cardinality of $\mathcal{C}_i^*$. Let $\mathcal{C}_{(z),i}^* \subseteq \mathcal{C}_i^*$ denote the set of data points of cluster $\mathcal{C}_i^*$ that are available on client z i.e., $\mathcal{C}_{(z)}^* = \cup\, \mathcal{C}_{(z),i}^*$. Suppose $n_i^{(z)}$ be the number of points in $\mathcal{C}_{(z),i}^*$. Also, say $X^{(z)}$ denotes the set of data points at client z. Further, we say that the complete set of data points X can be visualized as matrix $X \in \mathbb{R}^{n\times h}$ with the i^{th} row representing the data point $x_i \in X$. Let $[k]$ denote the set $\{1, 2, \ldots, k\}$. Further, suppose that $G^* \in \mathbb{R}^{n\times h}$ denotes a matrix such that each i^{th} row corresponds to the optimal center of data point $x_i \in X$. Similarly, let $G_{(z)}^*$ matrix be the corresponding G matrix but defined only for data points in $\mathcal{C}_{(z)}^*$ with centers defined on local datasets i.e., $\{c_{(z),1}^*, c_{(z),2}^*, \ldots, c_{(z),k}^*\}$.

Algorithm 12: Multishot Federated Clustering (`MFC`)

Client initially executes:

1 On each client $z \in [Z]$, run Algorithm 11 with local data $X^{(z)}$ and k to find local cluster centers $C^{(z)}$.

2 All clients $z \in [Z]$ shares center set $C^{(z)}$ with server.

Server initially executes:

1 Receives set of centers $C^{(z)}$ from all devices $z \in [Z]$, to construct $S = C^{(1)} \cup C^{(2)} \ldots \cup C^{(z)}$.

2 Apply Llyod k-means clustering [333] on set S, to find k global centers $C^g = \{c_1^g, c_2^g, \ldots, c_k^g\}$.

3 Sends back global centers C^g to all clients $z \in [Z]$ for further local training.

Client updates:

1 All clients $z \in [Z]$ receive global centers C^g from the server.

2 Each client z updates their local assignments function $\phi_{C^g}^{(z)}$ according to C^g, i.e., $\forall x_i \in \hat{X}^{(z)}$, $\phi_{C^g}^{(z)}(x_i) \leftarrow \text{argmin}_{c_j^g \in C^g} ||x_i - c_j^g||_2$

3 Updating local cluster sets $C^{(z)}$ by computing the mean of cluster assignments $\phi_{C^g}^{(z)}$.

4 Sends back local cluster center suffering maximum clustering cost (Definition 6.1) to server **(i.e., Server updates)**. Let us denote it using $c_{\max}^{(z)}$.

$$c_{\max}^{(z)} = \underset{c_j^{(z)} \in C^{(z)}}{\text{argmax}} \sum_{x_i \in X^{(z)}} ||x_i - c_j^{(z)}||_2^2$$

Server updates:

1 Receives maximum cost centers $c_{\max}^{(z)}$ from all $z \in [Z]$.

2 Update $S = S \cup \{c_{\max}^{(1)}, \ldots, c_{\max}^{(k)}\}$

3 Apply Llyod k-means clustering on the S, to find k global centers $C^g = \{c_1^g, c_2^g, \ldots, c_k^g\}$.

4 Sends back global centers C^g to all clients $z \in [Z]$ for further local training **(i.e., Client updates)**.

Assumption 1. *The non-empty subset of the data points on device z belonging to the global cluster $\mathcal{C}_i^*$, denoted by $\mathcal{C}_{(z),i}^*$ is sufficiently large. That is, there exists a sufficiently small constant $0 < \epsilon < 1$ such that $n_i^{(z)} \geq \frac{8\sigma_i^2}{\epsilon^2}\left(ln(\frac{1}{\delta}) + \frac{1}{4}\right)$ $\forall\ i \in [k]$ for a given $0 < \delta < 1$.*

Next, we define the notion of the well-separability of clusters, and such assumption is a standard in the clustering literature [109, 41].

Definition 6.3 (Well-separability)

A pair of target clusters $\mathcal{C}_i^*$ and $\mathcal{C}_j^*$ are said to be well separated if they satisfy

$$||c_i^* - c_j^*||_2 \geq \rho\sqrt{k}||X - G^*||_2\left(\frac{1}{\sqrt{n_i}} + \frac{1}{\sqrt{n_j}}\right)$$

where ρ is a large constant and n_i is number of data points in cluster $\mathcal{C}_i^*$.

Assumption 2. *The centers of the oracle clustering $\mathcal{C}_i^*$ are well separated.*

Having stated the assumptions that are well-adopted in centralized and federated clustering literature [41, 109]. We now look into some initial results that will eventually help bound the center separation in Lemma 6.3. In this section, we consider the distance metric as $p = 2$ norm (i.e., Euclidean distance), a popular metric in clustering literature [149]. The main reason is its symmetric nature and adherence to triangular inequality properties that help provide theoretical guarantees.

6.5.2 Theoretical Results

We first show that the centers obtained on local data points at each device (if optimal clustering would have been known) is close to that of global centers. For this, we use vector Bernstein inequality provided in Kohler and Lucchi [334]. We restate the lemma here for the sake of the completeness.

Lemma 6.1. *(Vector Bernstein Inequality [334]) Let $x_1, x_2, \ldots, x_n$ be h-dimensional independent vector-valued random variables such that $\mathbb{E}(x_a) = 0$, $||x_a||_2 \leq \mu$ and $\mathbb{E}\Big(||x_a||^2\Big) \leq \sigma^2$, then $\forall \epsilon : 0 < \epsilon < \frac{\sigma^2}{\mu}$, we have*

$$\mathbb{P}\left(\left|\left|\frac{1}{n}\sum_{a=1}^{n} x_a\right|\right|_2 \geq \epsilon\right) \leq exp\left(-n\frac{\epsilon^2}{8\sigma^2} + \frac{1}{4}\right)$$

Using the above lemma on a vector valued random variable $(x_j - c_i^*)$ where x_j is a data point present in local clustering $\mathcal{C}^*_{(z),i}$ where $i \in [k]$, we get:

Lemma 6.2. *Let σ_i^2 to be an upper bound on the variance of i^{th} global cluster i.e., $\mathbb{E}(||x_j - c_i^*||_2^2) \leq \sigma_i^2$. Now if $n_i^{(z)} \geq \frac{8\sigma_i^2}{\epsilon^2}\left(\ln(\frac{1}{\delta}) + \frac{1}{4}\right)$, then $||c^*_{(z),i} - c_i^*||_2 \leq \epsilon$ with probability at least $1 - \delta$.*

Proof. For completing the proof substitute $n_i^{(z)}$ in Lemma 6.1, we get, $exp\left(-n\frac{\epsilon^2}{8\sigma^2} + \frac{1}{4}\right) \leq \delta$. This implies

$$P\left(\left|\left|\frac{1}{n}\sum_{j=1}^{n}(x_j - c_i^*)\right|\right| \geq \epsilon\right) \leq \delta \implies P\left(\left|\left|\left(\frac{1}{n}\sum_{j=1}^{n} x_j\right) - c_i^*\right|\right| \geq \epsilon\right) \leq \delta$$

As in k-means, the local center is obtained by taking mean update of all the data points (here $x_j \in \mathcal{C}^*_{(z),i}$), so this implies, $\implies P(||c^*_{(z),i} - c_i^*|| \geq \epsilon) \leq \delta$. Hence proved. □

Now, further let $n_{max}^{(z)} = \max_{i\in[k]}\left(n_i^{(z)}\right)$ be the maximum number of data points in any cluster at client z. In order to have sufficiently small requirement of ϵ in Lemma 6.2 consider ω such that $\epsilon \leq \frac{\omega\sqrt{k}||X-G^*||_2}{\sqrt{n_{max}^{(z)}}} \leq \frac{\omega}{2}\sqrt{k}||X - G^*||_2\left(\frac{1}{\sqrt{n_i^{(z)}}} + \frac{1}{\sqrt{n_j^{(z)}}}\right)$. Here $i, j \in [k]$ denotes any two clusters on client z.

We now next show that if the original (optimal) true distribution centers are well separated, then the optimal centers computed using local data points available at clients will also be well separated. This lemma uses the Lemma 6.2 and will help us use the next lemma proved by Awasthi and Sheffet [109].

Lemma 6.3. *Given* $\gamma = \frac{8\sigma_i^2}{n_i\epsilon^2}\left(\ln\left(\frac{1}{\delta}\right) + \frac{1}{4}\right)$ *and well-separability for each pair of centers in true optimal clustering (or say optimal global cluster centers) we say that for some* ρ' *chosen such that* $\rho' \geq (\rho\sqrt{\gamma} - \omega)$ *we will have well-separability for optimal centers on local data for all clients hold true with probability at least* $1-\delta$*. That is for any cluster* $i, j \in [k]$ *on client* z *we have,*

$$P\left(||c^*_{(z),i} - c^*_{(z),j}||_2 \geq \rho'\sqrt{k}||X - G^*||_2\left(\frac{1}{\sqrt{n_i}} + \frac{1}{\sqrt{n_j}}\right)\right) \geq (1-\delta)$$

Proof. Starting with left-hand side and using triangular inequality along with Lemma 6.2 we get,

$$\begin{aligned} ||c^*_i - c^*_j||_2 &\leq ||c^*_{(z),i} - c^*_i||_2 + ||c^*_{(z),j} - c^*_j||_2 \\ &\quad + ||c^*_{(z),i} - c^*_{(z),j}||_2 \\ &\leq 2\epsilon + ||c^*_{(z),i} - c^*_{(z),j})||_2 \end{aligned} \tag{6.4}$$

Now since for any two cluster number $i, j \in [k]$ on client z we have $\epsilon \leq \frac{\omega}{2}\sqrt{k}||X - G^*||_2\left(\frac{1}{\sqrt{n_i^{(z)}}} + \frac{1}{\sqrt{n_j^{(z)}}}\right)$, and as each pair of global clusters are well separated, we have

$$||c^*_{(z),i} - c^*_{(z),j}||_2 \geq \rho\sqrt{k}||X - G^*||_2\left(\frac{1}{\sqrt{n_i}} + \frac{1}{\sqrt{n_j}}\right) - 2\epsilon \tag{6.5}$$

$$||c^*_{(z),i} - c^*_{(z),j}||_2 \geq \rho\sqrt{k}||X - G^*||_2\sqrt{\gamma}\left(\frac{1}{\sqrt{n_i^{(z)}}} + \frac{1}{\sqrt{n_j^{(z)}}}\right) - 2\epsilon \tag{6.6}$$

$$||c^*_{(z),i} - c^*_{(z),j}||_2 \geq (\rho\sqrt{\gamma} - \omega)\sqrt{k}||X - G^*||_2\left(\frac{1}{\sqrt{n_i^{(z)}}} + \frac{1}{\sqrt{n_j^{(z)}}}\right) \tag{6.7}$$

$$||c^*_{(z),i} - c^*_{(z),j}||_2 \geq \rho'\sqrt{k}||X - G^*||_2\left(\frac{1}{\sqrt{n_i^{(z)}}} + \frac{1}{\sqrt{n_j^{(z)}}}\right) \tag{6.8}$$

□

We now restate the result of Awasthi and Sheffet [109] about guarantees on the quality of centers obtained by applying their procedure described in Algorithm 11.

Lemma 6.4 (Awasthi and Sheffet [109])**.** *If each pair of centers obtained on local data are well separated, then after performing Algorithm 11 for obtained centers* $\{c_i^{(z)}\}_{i=1}^k$ *for*

any client $z \in Z$, we have,

$$||c_i^{(z)} - c^*_{(z),i})||_2 \leq \frac{25}{\rho'} \frac{||X^{(z)} - G^*_{(z)}||_2}{\sqrt{n_i^{(z)}}}$$

It should be noted that the above holds only when well-separability is held. Specifically, in Algorithm 11, we create k clusters, while in reality, the data points on a device may come from only $k_z (\leq k)$ clusters. But, we can observe that if the well-separability is violated for any pair of centers then it will only occur if these centers belong to the same true distributions (or optimal clusters). In fact, such centers, if obtained, will be close to the center that would have been obtained if $k = k_z$. This can be proved with the help of the central limit theorem by considering the violating pair of centers to be two different sample means of the same population (i.e. target global cluster). To account for this, we introduce a mapping $\Gamma^{(z)} : [k] \to [k]$ which essentially maps the index of the local cluster center obtained by Algorithm 11 to the index of the closest global center. Thus, we have $\Gamma^{(z)}(j) = \text{argmin}_{i \in [k]} ||c_j^{(z)} - c^*_{(z),i}||_2$. Thus, as a direct consequence of Lemma 6.4, we get

$$||c_j^{(z)} - c^*_{(z),\Gamma^{(z)}(j)}||_2 \leq \frac{25}{\rho'} \frac{||X^{(z)} - G^*_{(z)}||}{\sqrt{n^{(z)}_{\Gamma^{(z)}(j)}}} \tag{6.9}$$

We now restate the result provided in Dennis et al. [41], which provides an upper bound on the cost when using global centers on local data in terms of the total optimal cost on global centers. These lemmas will help us prove the main Theorem 6.6.

Lemma 6.5 (Dennis et al. [41])**.** *Given k as number of clusters and $X^{(z)}$ as local data with $G^*_{(z)}$ as optimal centers. Then we have, $||X^{(z)} - G^*_{(z)}||_2 \leq 2\sqrt{k}||X - G^*_{(z)}||_2$*

Theorem 6.6 (Main Theorem)**.** *Given X as set of data points sampled from k true distributions with G^* as set of optimal centers. We say that if well separability holds with high probability, then for every local center (i.e., $c_j^{(z)} \ \ \forall j \in [k]$) computed using `MFC` at any client $z \in Z$ is close to the corresponding optimal centers. That is*

$$||c_j^{(z)} - c^*_{\Gamma^{(z)}(j)}||_2 \leq c\sqrt{k} \frac{||X - G^*||_2}{\sqrt{n^{(z)}_{\Gamma^{(z)}(j)}}},$$

Here c is a positive constant and Γ is the mapping of the local cluster center to the closest optimal (or global) center.

Proof. We prove this theorem in two steps. Firstly, using Equation 6.9 and Lemma 6.5, we have

$$||c_j^{(z)} - c^*_{\Gamma^{(z)}(j)}||_2 \leq \frac{50\sqrt{k}}{\rho'} \frac{||X - G^*||_2}{\sqrt{n^{(z)}_{\Gamma^{(z)}(j)}}} \tag{6.10}$$

Now, $||c_j^{(z)} - c^*_{\Gamma^{(z)}(j)}||_2 \leq ||c_j^{(z)} - c^*_{(z),\Gamma^{(z)}(j)}||_2 + ||c^*_{(z),\Gamma^{(z)}(j)} - c^*_{\Gamma^{(z)}(j)}||_2$ (using triangular inequality)

$$\leq \frac{50\sqrt{k}}{\rho'} \frac{||X - G^*||_2}{\sqrt{n^{(z)}_{\Gamma^{(z)}(j)}}} + \epsilon \qquad \text{(using Equation 6.10 and value of } \epsilon\text{)}$$

$$\leq \frac{50\sqrt{k}}{\rho'} \frac{||X - G^*||_2}{\sqrt{n^{(z)}_{\Gamma^{(z)}(j)}}} + \frac{\omega\sqrt{k}}{1} \frac{||X - G^*||_2}{\sqrt{n^{(z)}_{\Gamma^{(z)}(j)}}} + \epsilon \qquad (\because n^{(z)}_{max} \geq n^{(z)}_{\Gamma^{(z)}(j)})$$

So if $c = \frac{50}{\rho'} + \omega$, we get the required result. □

Beyond this, the multi-shot iterations only perform a Lloyd's heuristic; thus, centers only improve and help further lower this bound on distance. Therefore, from the above theorem, the `MFC` converges after a few shots. Determining the exact number of shots (or communication rounds) is challenging as it depends on the nature of the dataset and the distribution of data points across clients. We leave this analysis as future work and conjecture the following:

Lemma 6.7. *`MFC` algorithm determines centers close to optimal clustering and converges after a few shots.*

Note that in `MFC`, all the clients share all the k centers with the server. We will now describe a more privacy-preserving modification of `MFC` that shares only one center. Though this modification offers communication efficiency and privacy, it demands that a sufficient number of clients should be available. We now provide bounds on a minimum number of clients required to achieve similar performance guarantees as when clients in `MFC` shared all k centers with the server in the initialization round. More specifically, we show that if enough clients are available, the server will have at least one representation from each Gaussian distribution, even when clients share back only one center. With this, one can simply follow the previous results (Theorem 6.6) to bound the closeness of global and the obtained local centers.

Lemma 6.8. *Given that there are at least $O(k^2 \log(k))$ clients available in the federated system, then after the first round of communication, the server will receive at least one data point (or representation) from each of the true k Gaussian distributions.*

Proof. Our goal is to determine the minimum expected number of clients that needs to be included in the network to ensure the representation of data points from all k Gaussian's. We can map this problem to the classical probabilistic problem called the coupon collector problem [335]. In the coupon collector problem, there are a total of k different types of coupons, and if each coupon type is arriving uniformly at random, we need to find the expected number of purchases that are needed to collect at least one coupon of type. More formally, it is defined as:

Claim 6.9 (Non Uniform Coupon Collector Problem [259])**.** *Given m distinct coupon types, the expected number of coupons required to obtain at least one coupon from each type is denoted as $\mathcal{H}_m$, and it is calculated as follows: $\mathcal{H}_m = \sum_{a=1}^{m} (-1)^{a-1} \sum_{1 \leq j_1, \ldots, j_a \leq k} \frac{1}{p(j_1)+\ldots+p(j_a)}$ where $p(i)$ is the probability of obtaining a coupon of type i.*

We will now apply the principles of the coupon collector problem to solve our problem. However, the challenge lies in computing the probabilities of each coupon type. We first discuss the case when data points are uniformly distributed across clients (i.e., $H_m = m \log(m)$) and treat the k Gaussian distributions as k coupon type ($m = k$). Now, since clients may have data points from fewer than k distributions, specifically $k_z \leq k$. To handle this, we first determine the minimum number of devices needed to represent data points from a particular distribution, say (i^{th} distribution) on the server. Let us say that we need m clients that have data points from i^{th} distribution on them. Applying the coupon collector problem, we can say that if we have at least $m = k_z \log k_z \leq k \log k$ clients, then there will be at least one data point (or representation) from i^{th} distribution at the server. To extend the bound to all k distributions, we can, along similar lines, say that if we have $\sum_{i=1}^{k} k \log k = k^2 \log k$ devices, then all the distributions will be represented by modified `MFC`. With this bound, all the lemmas hold valid as previously. If prior information about distribution probability is known, one can use Claim 6.9 on same lines to find the minimum number of clients required. □

Though `MFC` helps overcome the challenge of data distribution dependence and, under well-separability assumptions, has theoretical bounds on the gap between obtained and optimal centers. Still, in both `k-FED` and `MFC` from our experimentation (Section 6.7), we observe that these methods can have high-cost deviations across clients. Though `MFC` performs much better than `k-FED` in this regard, there is still room for improvement. Thus, we extend `MFC` to incorporate the idea of personalization for each client. The proposed method called `p-FClus` ensures lower cost deviation across clients. It helps clients tune their centers according to their local needs, as desired in our banking example in Section 6.1. Personalization has been adopted quite well in supervised FL literature [271, 336], but we are the first to incorporate these ideas into FedUL. In the next section, we provide the primary working mechanism behind the algorithm that helps solve the non-trivial extension of achieving personalization (see Section 6.1) from supervised literature to unsupervised federated data clustering.

6.6 `p-FClus`: personalized Federated Clustering Algorithm

We propose a novel algorithm called `p-FClus` (**p**ersonalized-**F**ederated **Clus**tering). The algorithm mainly comprise of three phases, which are explained in subsequent subsections below. The complete pseudo-code for `p-FClus` is described in Algorithm 13, and its

implementation is available on the public repository[4].

Algorithm 13: `p-FClus`$(X^{(z)}\}_{z=1}^{Z}, k, p, \eta, \lambda)$

$\forall\ z \in [Z]$ in parallel:
  $C^{(z)} \leftarrow$ **clientInitialization**$(X^{(z)}, k, p)$
call procedure **Server**$\left(\left\{C^{(z)}\right\}_{z=1}^{Z}\right)$
/* Each client receives set of global cluster centers C^g */
$\forall\ z \in [Z]$ in parallel:
  /* Each client personalizes the C^g using stochastic gradient descent with learning rate η and fine-tuning level λ */
  $C^{(z)}, \phi^{(z)} \leftarrow$call procedure **clientPersonalization**$\left(C^{(z)}, C^g, \eta, \lambda, X^{(z)}\right)$
/* Global cluster centers received at all clients C^g from server has been personalized (fine-tuned) for use.*/
return $\left\{\phi^{(z)}, C^{(z)} : C^g \text{ (personalized) }\right\}_{z=1}^{Z}$

Algorithm 14: `p-FClus`'s Client-side Initialization Procedure$(X^{(z)}, k, p)$

/* Apply (k, p)- clustering using Lloyd [333] for $p = 2$ (k-means) and Charikar et al. [337] for $p = 1$ (k-mediod). */
$C^{(z)} \leftarrow$ **(k, p)-clustering**$(X^{(z)})$
return $C^{(z)}$

Algorithm 15: `p-FClus`'s Server-side Procedure$(k, p, \left\{C^{(z)}\right\}_{z=1}^{Z})$

/*Post client initialization*/
$S \leftarrow \cup_{z\in[Z]} C^{(z)}$
/*Apply (k, p)-clustering Lloyd [333], Charikar et al. [337] for p value of $2, 1$ respectively) on S to get k global centers*/
$C^g \leftarrow$ **(k, p)-clustering**(S)
In parallel $\forall z \in [Z]$: **return** C^g to all clients

6.6.1 Client Initialization

Initially, all the clients in parallel run an initialization procedure as described in Algorithm 14. Primarily, each client $z \in [Z]$ executes a p-norm clustering algorithm to find a set of k local cluster centers ($C^{(z)}$). These centers can be computed using known heuristics or approximation algorithms, such as those described in Lloyd [333] for p-norm value of two (k-means) and in Charikar et al. [337] for the k-medoid objective ($p = 1$). These methods ensure that the centers obtained minimize the objective cost on the local datasets. After computing their respective local cluster centers, each client shares its set of local cluster centers $C^{(z)}$ with the server. The task then shifts to sever, which then executes the server-side procedure discussed in the following subsection.

[4]https://github.com/P-FClus/p-FClus

6.6.2 Server Execution

Algorithm 15 summarises the complete server-side procedure. In line number 1 to 2 of procedure, after receiving the set of centers $C^{(z)}$ from all devices $z \in [Z]$, the server constructs a set S by aggregating them, i.e., $S = C^{(1)} \cup C^{(2)} \cup \ldots \cup C^{(z)}$. Subsequently, the server applies (k, p)-clustering algorithm [333, 337] to this set S to determine set of k global centers C^g in line 3. These global centers are expected to minimize the objective cost across clients. However, in some cases, these centers can still be far from the current local data due to heterogeneity. So these global centers are distributed back to all clients and thereafter undergo personalization to find centers with lower costs and, therefore, lower cost deviation across clients..

6.6.3 Client Side Personalization

After receiving the set of global centers, all clients use their set of local centers to fine-tune the global centers to form personalized centers using procedure available in Algorithm 16. The fine-tuning level λ can be kept consistent across clients, or clients can vary it according to their preferences. For each h-dimensional data point in $x_i \in X^{(z)}$, the client identifies the closest local ($c^{(z)} \in C^{(z)}$, line number 5) and global center vector ($c^g \in C^g$, line number 6) and fine-tunes the global ones by minimizing the following function for finite p-norm:

$$P(x) = \underbrace{\frac{1}{2}\left\|c^g - x\right\|_p}_{\text{clustering cost}} + \underbrace{\lambda\left(c^g - c^{(z)}\right)^2}_{\text{regularization penalty}} \quad \text{(Personalization Objective)} \tag{6.11}$$

The above personalization objective (Equation 6.11) emphasizes updating global centers to minimize the local cost ($L_p^{(z)}(C^g)$, Definition 6.1) while ensuring that C^g does not collapse to $C^{(z)}$ by addition of a regularization factor. In other words, the role of the regularization factor is to ensure that the global centers are not too much deviated by incorporating penalty terms in the form of L_2-regularization.

Algorithm 16: `p-FClus`'s Client Personalization procedure $\left(C^{(z)}, C^g, \eta, \lambda, X^{(z)}\right)$

1: Initialize assignment function $\phi^{(z)} \leftarrow \Phi$ (empty)
2: **while** tuning steps t or convergence **do**
3: $t \leftarrow t - 1$
4: **for** $x \in X^{(z)}$ **do**
5: $c^{(z)} \leftarrow \text{argmin}_{c^{(z)} \in C^{(z)}} \left(d(x, c^{(z)})\right)$
6: $c^g \leftarrow \text{argmin}_{c^g \in C^g} \left(d(x, c^g)\right)$
7: $\eta \leftarrow 1 \Big/ \left|\mathbb{I}\left(\text{argmin}_{c^g \in C^g}\, d(x, c^g) = c^g\right)\right|$
8: $c^g \leftarrow c^g - \eta\Delta_{c^{(g)}}\left(P(x)\right)$ (Using Equation 6.12 or 6.13 and λ) /*C^g are personalized centers for client z*/
9: $\phi^{(z)}[x] \leftarrow c^g$
10: **return** $C^g, \phi^{(z)}$

Now, for k-means, the norm (p) takes the value of two and since minimizing euclidean

distance (2-norm) is the same as minimizing squared euclidean, therefore the derivative with respect to global center results in the following:

$$\Delta_{c^g}(P(x))\Big]_{p=2} = (c^g - x) + 2\lambda(c^g - c^{(z)}) \tag{6.12}$$

In the case of k-medoids, the norm takes the value of one i.e., minimizes 1-norm distance. Therefore, we get the derivative value as follows:

$$\Delta_{c^g}(P(x))\Big]_{p=1} = 1/2 + 2\lambda(c^g - c^{(z)}) \tag{6.13}$$

Now, we can update the global center using the Stochastic Gradient Descent (SGD) based personalization objective and find the updated global center that is not too far from local ones in line number 8 of the procedure.

$$c^g \leftarrow c^g - \eta\Delta_{c^g}(P(x)) \tag{6.14}$$

where we set $\eta = \left|\frac{1}{\mathbb{I}\left(\text{argmin}_{c^g \in C^g} d(x,c^g)=c^g\right)}\right|$ i.e, multiplicative inverse of the number of data points currently assigned to c^g act as learning rate (line number 7).

Note that in the special case of the 1-norm, i.e., k-medoids, there is a constraint that the center should be a data point. However, during the personalization process, it can happen that the final obtained center may not be a data point. In such cases for k-mediod objective, the nearest data point to the final center within the client's local data is chosen as the center.

Convergence of `p-FClus`: Note that as `p-FClus` initially uses local clustering methods [333, 337] already proven to converge and then applies Lloyd [333] that converges in most real-world scenarios. After these steps, the personalization procedure considers each data point only once. Thus, we can say that `p-FClus` converges.

Novelty: Therefore, to summarize, it is important to note that the non-trivial nature of extending the literature in supervised personalization [271, 336] to unsupervised learning is handled by the intrinsic design of the `p-FClus` procedure. Rather than demanding the need for having a synchronization (or chronological) ordering on centers across clients for direct averaging, unlike prior works, we use the data point-wise gradient update and use nearest local, global center mappings. We now validate the efficacy of the `p-FClus`, `MFC` against state-of-the-art (`SOTA`) methods.

6.7 Experimental Result and Analysis

We will now validate the performance of the proposed `p-FClus`[5] and `MFC` against `SOTA` approaches on different synthetic and *benchmarking* real-world datasets used in clustering literature [42, 12, 338, 10]. These are as follows:

[5]https://github.com/P-FClus/p-FClus

- **Synthetic Datasets** (`Syn`) - The synthetic datasets generated can mainly be categorized into the following:
  - **No Overlap** (`Syn-NO`): It contains data points from ten bi-variate gaussian distributions $\{\mathcal{N}_i(10i, 1)\}_{i=1}^{10}$, ensuring that the dataset has well-separated clusters.
  - **Little Overlap** (`Syn-LO`): It consists of data points from ten bi-variate gaussian distributions arranged in a way that the consecutive pairs of distributions overlap only after two standard deviations. The standard deviation is set to two for all gaussian's.
  - **Overlapping** (`Syn-O`): It consists of data points from ten bi-variate gaussian distributions arranged such that in each consecutive pair of distributions, the mean of one distribution and three standard deviations of the other distribution in the pair touch each other. The standard deviation is set to three for all gaussian's. The data generation code is available in the code repository[6].
- **Real-world Datasets** - The real-world datasets used in the study can be further divided into two types. The first type comprises of datasets that require pre-processing to make them ready for use in a federated environment i.e, they are extrinsic in nature. The second type includes datasets that are inherently captured in a federated manner, where the data points are naturally divided among clients.
  - Non-Federated Datasets (Extrinsic)
    - **`Adult`**: The census record collection of 1994 US citizens. It comprises 32562 records with feature attributes under present study as age, fnlwgt, education_num, capital_gain, and hours_per_week. These attributes are consistent with prior works on clustering [10]. The dataset is openly available[6].
    - **`Bank`**: A direct phone call marketing campaign data of banks in the Portugal region. It comprises 41108 records containing information about consumers' age, duration, campaign, cons.price.idx, euribor3m, nr.employed as attributes. The features selected for experimentation align with previous literature, and the dataset is publicly available[7].
    - **`Diabetes`**: US clinical records collected over ten years. The features chosen are age and time_in_hospital, and is publicly available[8].
    - **`FMNIST`**: Contains $60,000$ training images covering ten classes of fashion items, each at a resolution of 28×28 pixels and is publicly accessible[9].
  - Federated Datasets (Intrinsic)

[6]https://archive.ics.uci.edu/ml/datasets/Adult
[7]https://archive.ics.uci.edu/ml/datasets/Bank+Marketing
[8]https://archive.ics.uci.edu/ml/datasets/Diabetes+130-US+hospitals+for+years+1999-2008
[9]https://github.com/zalandoresearch/fashion-mnist/

* **FEMNIST**: It is a handwritten character recognition dataset where each client corresponds to a writer from the EMNIST dataset[10]. The dataset's intrinsic heterogeneity is $\mathtt{H} = 62$, i.e., data points from 62 true distributions are distributed among 500 clients during data collection.
* **WISDM**: It is a publicly available wireless sensor data mining dataset consisting of $1,098,207$ samples for activity recognition using mobile phone accelerometers[11] for recognizing six ($\mathtt{H} = 6$) activities: walking, jogging, going upstairs, going downstairs, sitting, and standing across 36 clients.

We compare `p-FClus`, `MFC` on both k-means (p-norm of 2) and k-mediod (p-norm of 1) objectives against the following baselines:

- **Centralized Clustering** (`CentClus`) [228]: An euclidean-based centralized version of k-centroid clustering. The aim is to minimize the objective cost using a norm value of two ($\ell = 2$), resulting in the well-known Lloyd's heuristic (or simply k-means) [333]. When the norm value is one ($\ell = 1$), a centralized variation of k-mediod clustering is achieved, where the set of centers is restricted to the data points in the dataset [337].
- **Oneshot Federated Clustering** (`k-FED`) [41]: The method is a federated data clustering approach that leverages the data heterogeneity among clients. The method executes Awasthi's k-means [109] locally on clients, and then clients share information about local centers with the server. The server then applies a variant of the farthest heuristic to select the best k global centers. These global centers are then shared back with clients for local clustering. Note that the method works well only when the network has high heterogeneity.

An important point to note is that the `k-FED` and `MFC` methods are intrinsically designed to work only for the k-means objective. Thus, this limits comparing these methods to k-means version of our `p-FClus`. To validate the performance of the k-median objective of `p-FClus`, we compare it to the centralized setting. The metrics involved in comparing the efficacy include the following:

- **Mean Cost per data point** ($\boldsymbol{\mu} \downarrow$): It is the mean (or average) objective cost experienced by each of the data points across clients and is lower the better. It is computed as described in Equation 6.2.
- **Cost Deviation per data point** ($\boldsymbol{\sigma} \downarrow$): It is a fairness metric that measures the standard deviation in per-point cost experienced by clients. The empirical value is estimated using Equation 6.3.
- **Maximum Cost per data point** (**max** $\downarrow$): It helps in estimating the worst per point cost that any client has to suffer and is computed as $\mathbf{max} = \max_{z \in [Z]} \boldsymbol{\mu}^{(z)}(C)$. A lower value indicates a more fairer clustering for clients.

[10] https://github.com/TalwalkarLab/leaf/tree/master/data/femnist
[11] https://www.cis.fordham.edu/wisdm/dataset.php

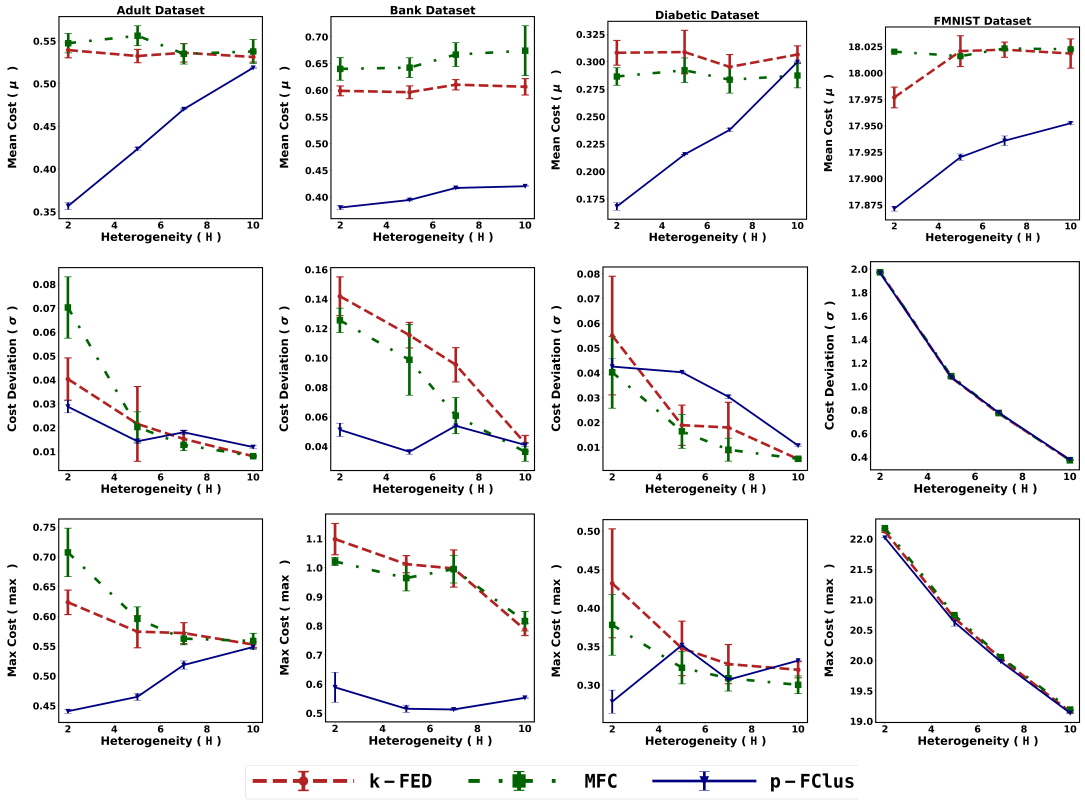


Figure 6.1: The plot shows the variation in evaluation metrics for proposed `p-FClus`, `MFC` and `SOTA` on k-means for varying heterogeneity levels on a `Balanced` data split across 100 clients. Each column represents a dataset as specified at the top, and each row represents one metric under evaluation. Note that the `FMNIST` dataset is on 500 clients. (Best viewed in color).

Note that we consider C in all these metrics as personalized global centers for `p-FClus`, and for `k-FED`, `MFC`, we consider C as the set of global centers computed by the method.

6.7.1 Experimental Setup

All experiments are conducted on an Intel Xeon $6246R$ processor with 280GB of RAM, running Ubuntu 18 and Python 3.8. We report the results as the mean and standard deviation of five independent runs, with the seed chosen from the set $\{0, 300, 600, 900, 1200\}$. The complete reproducible code is available online[6]. Next, we investigate the distribution of data among clients, focusing mainly on two different settings described below:

1. `Balanced` or (Equal) distribution: In this setting, random data points from each true distribution ($\mathcal{D}_k$) are equally divided among clients. Note that in scenarios where the total number of data points from $\mathcal{D}_k$ is not divisible by the number of clients, each client will have nearly equal data points or may have one less data point compared to other clients.

2. `Unequal` distribution: In this setting, clients can have a different number of data

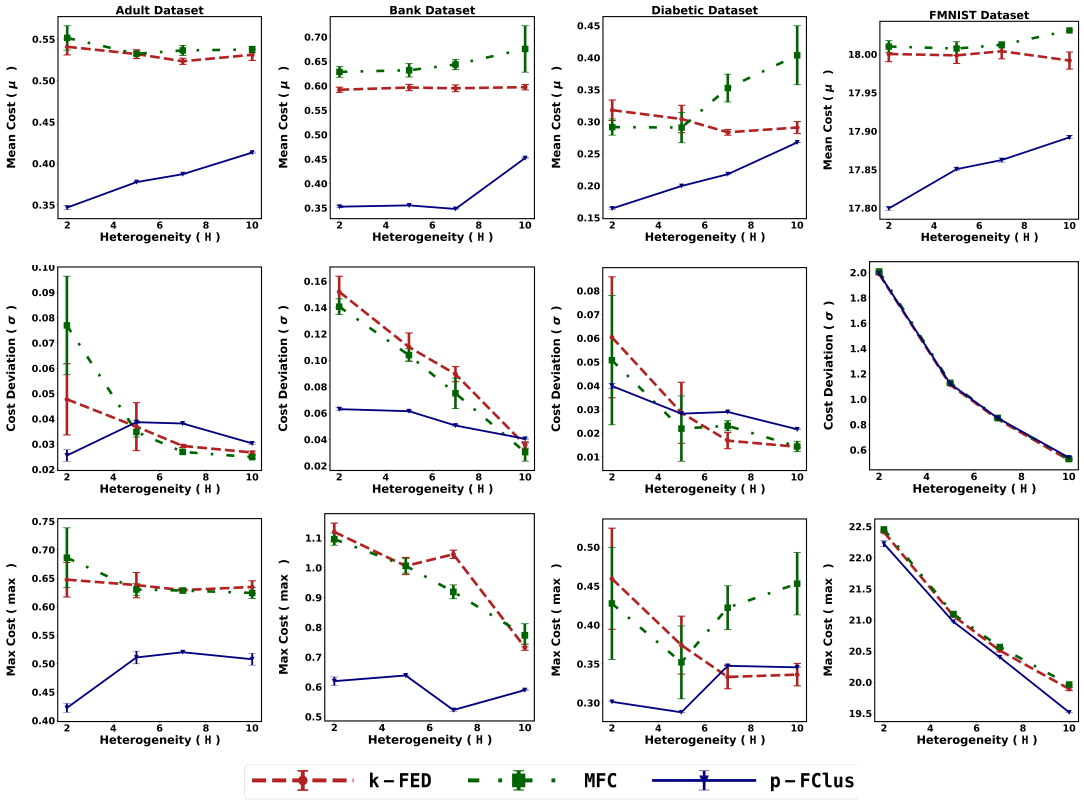


Figure 6.2: The plot shows the variation in evaluation metrics for proposed `p-FClus`, `MFC` and `SOTA` on k-means objective for varying heterogeneity levels on a `Balanced` data split across 1000 clients. Each column represents a specific dataset as specified at the top, and each row represents one metric under evaluation. (Best viewed in color).

points from each of the $\mathcal{D}_k$ distributions. Instead of arbitrarily dividing the data points among clients, we intelligently distribute the data points to capture scenarios where some clients may have significantly fewer data points from certain D_k, resulting in an overall skewed division among clients. The code repository contains scripts for generating the same[6].

Within each of the above settings, we further consider the level of heterogeneity (`H`) as $2, 5, 7$ and 10 (Here, 10 is the maximum number of distributions in non-federated datasets under consideration [44]). This implies that if, for instance, the `H` $= 5$, then every client will contain data points only from any of the 5 ($\leq k$) distributions.
We now begin by validating our `p-FClus`,`MFC` against `SOTA` on a `Balanced` distribution setting and non-federated datasets. Later, we will explore the scenario where clients can have an `Unequal` data distribution and then on intrinsic or fixed heterogeneity federated datasets (`FEMNIST` and `WISDM`).

6.7.2 Analysis on `Balanced` Data Distribution among Clients on k-means Objective

This subsection delves into the results of the balanced (or equal) data distribution setting in the k-means objective (p-norm $= 2$). The results are illustrated in Figure 6.1 (Real-world

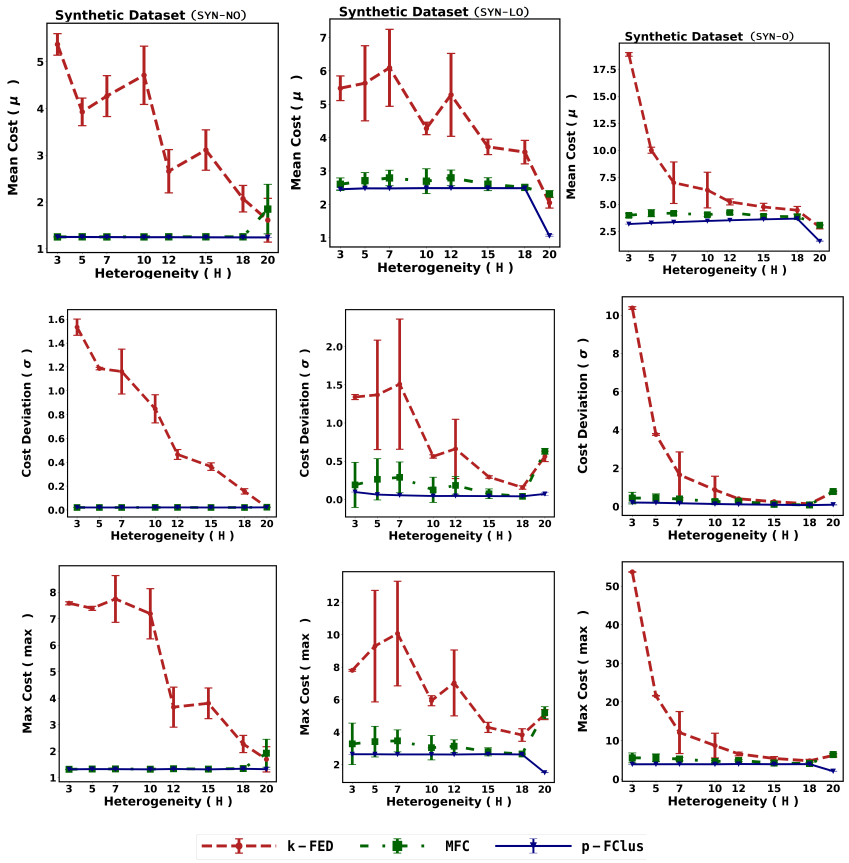


Figure 6.3: The plot shows the variation in evaluation metrics for proposed `p-FClus`, `MFC` and `SOTA` on k-means objective for varying heterogeneity levels on a `Balanced` data split across 1000 clients. Each column represents a specific Synthetic dataset (`Syn`) in sequence: `Syn-NO`, `Syn-LO`, `Syn-O` respectively, and each row represents one metric under evaluation. (Best viewed in color).

dataset, 100 clients), Figure 6.2 (Real-world dataset, 1000 clients) and Figure 6.3 (`Syn` dataset). We first provide a brief overview of the observations for each dataset in the subsections below and then conclude the overall results in this setting.

Observations for `Adult`

It can be observed from Figure 6.1 and Figure 6.2 that for both 100 and 1000 clients we have the mean per-point cost ($\boldsymbol{\mu}$) for `p-FClus` significantly lower than that of `SOTA`, especially in more challenging settings of lower heterogeneity. Furthermore, the variance of $\boldsymbol{\mu}$ is lower compared to both `k-FED` and `MFC`, showing the efficacy of `p-FClus` in achieving a lower objective cost for all number of clients settings. Additionally, the fairness metric $\boldsymbol{\sigma}$ for `k-FED` and `MFC` is higher at lower levels. In contrast, `p-FClus` helps achieve a lower $\boldsymbol{\mu}$ for all clients and stays within a fixed confidence region by fine-tuning using local data to bring the global centers close to local ones without deviating significantly from the global model. This shows that `p-FClus` is not sensitive to changes in heterogeneity levels. Approaches such as `k-FED` and `MFC` can achieve better $\boldsymbol{\mu}$ and $\boldsymbol{\sigma}$ at higher heterogeneity, as in such scenarios, they can capture good estimates of global centers mainly due to sufficient

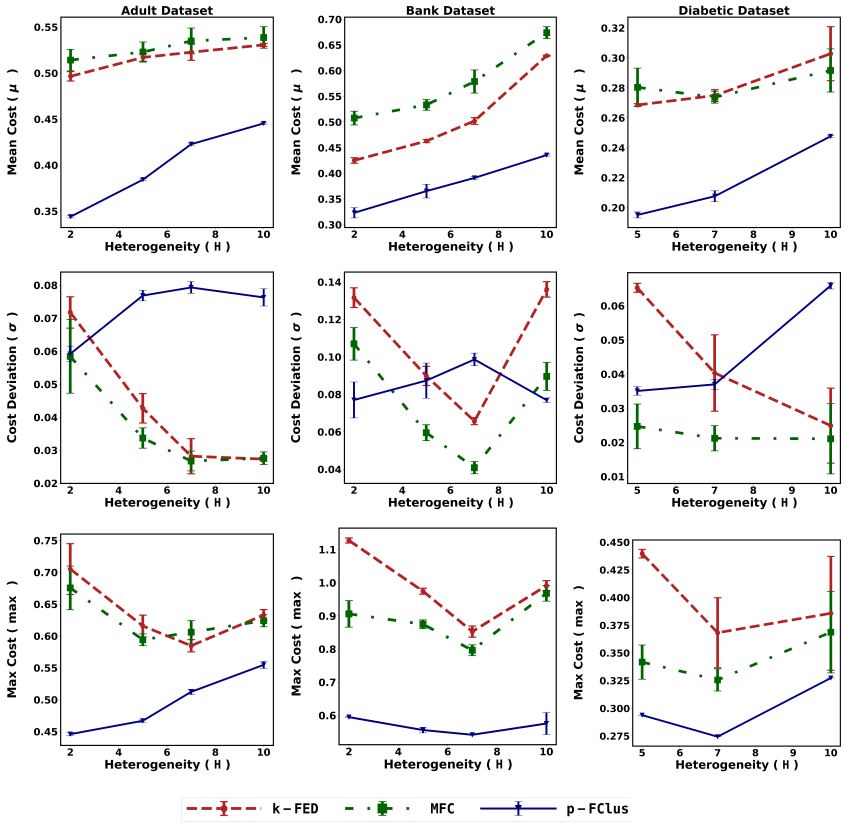


Figure 6.4: The plot shows the variation in evaluation metrics for proposed `p-FClus`, `MFC` and `SOTA` on k-means objective for varying heterogeneity levels on a `Unequal` data split across 100 clients. Each column represents a specific dataset as specified at the top, and each row represents one metric under evaluation. (Best viewed in color).

data availability from all distributions across clients. Notably, the maximum per-point cost (**max**) remains consistently high for `SOTA` methods compared to `p-FClus`, showcasing the presence of clients that might be willing to not contribute and leave the federated system. Though there is an increasing trend for `p-FClus`, it remains at a significant gap from `SOTA`. Therefore, in the `Adult` dataset, `p-FClus` performs considerably well.

Observations for `Bank`

The observations are quite aligned with the `Adult` dataset. $\boldsymbol{\mu}$, **max** is considerably lower than `SOTA` even on varying heterogeneity (`H`) and number of clients. Also, the fairness metric, $\boldsymbol{\sigma}$ for `p-FClus` remains below `SOTA` for most heterogeneity (`H`) levels.

Observations for `Diabetes`

The additional comments to observe is that though in `Diabetes` dataset `k-FED` and `MFC` have lower $\boldsymbol{\sigma}$ with `MFC` having the least value. But both these methods suffer high variance compared to `p-FClus`. This is primarily due to local optima in the dataset [44]. Also, the **max** for `p-FClus` is either lower or comparable to `SOTA` methods, showcasing the robustness of `p-FClus` to local optima's.

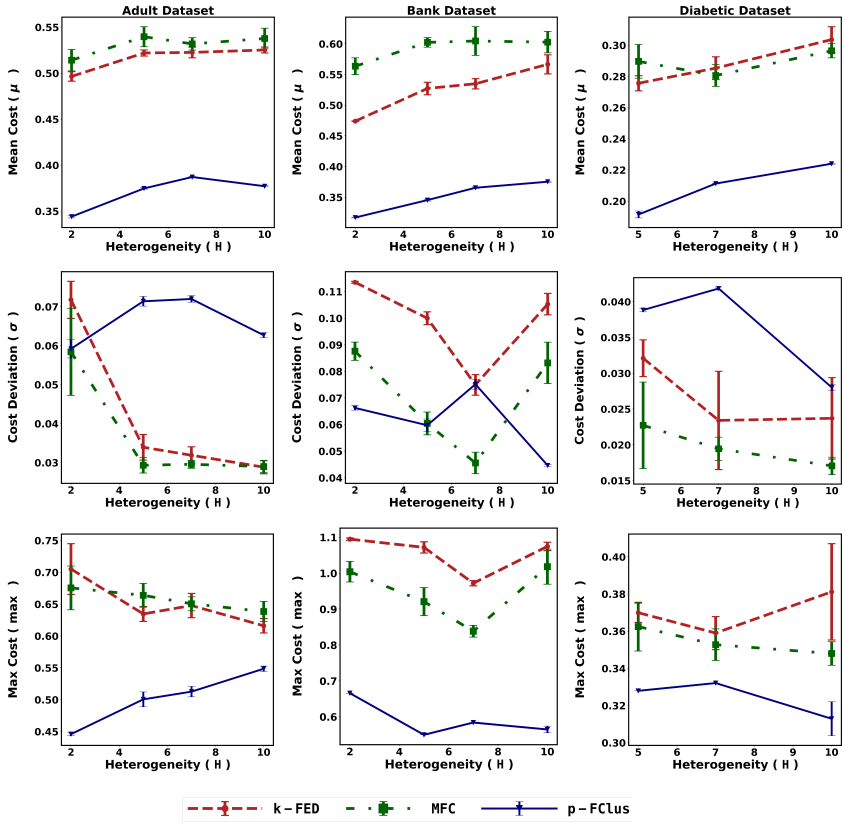


Figure 6.5: The plot shows the variation in evaluation metrics for proposed `p-FClus`, `MFC` and `SOTA` on k-means Objective for varying heterogeneity levels on a `Unequal` data split across 500 clients. Each column represents a specific dataset as specified at the top, and each row represents one metric under evaluation. (Best viewed in color).

Observations for `FMNIST`

The $\boldsymbol{\sigma}$ of all approaches are quite close enough, but `p-FClus` has considerably better performance on other metrics, namely $\boldsymbol{\mu}$ and **max**.

Observations for Synthetic (`Syn`)

We can observe from the Figure 6.3 that when the dataset is bearing no overlaps between different clusters, both `MFC` and `p-FClus` have comparable costs, but as the overlap increases in `Syn-LO` and `Syn-O`, the `MFC` method slightly deviates and achieves higher mean per point cost ($\boldsymbol{\mu}$) and higher fairness metrics i.e., deviation ($\boldsymbol{\sigma}$) and maximum cost (**max**). On the other hand, across all different settings, `k-FED` always exhibits significantly poorer performance in terms of mean cost and fairness metrics.

Overall Insight: In a `Balanced` data distribution, `p-FClus` achieves a lower per-point cost ($\boldsymbol{\mu}$, owing to personalization) and a more fair solution ($\boldsymbol{\sigma}$) across clients, making it a reliable choice when information about the level of heterogeneity in the network is unknown or unstable.

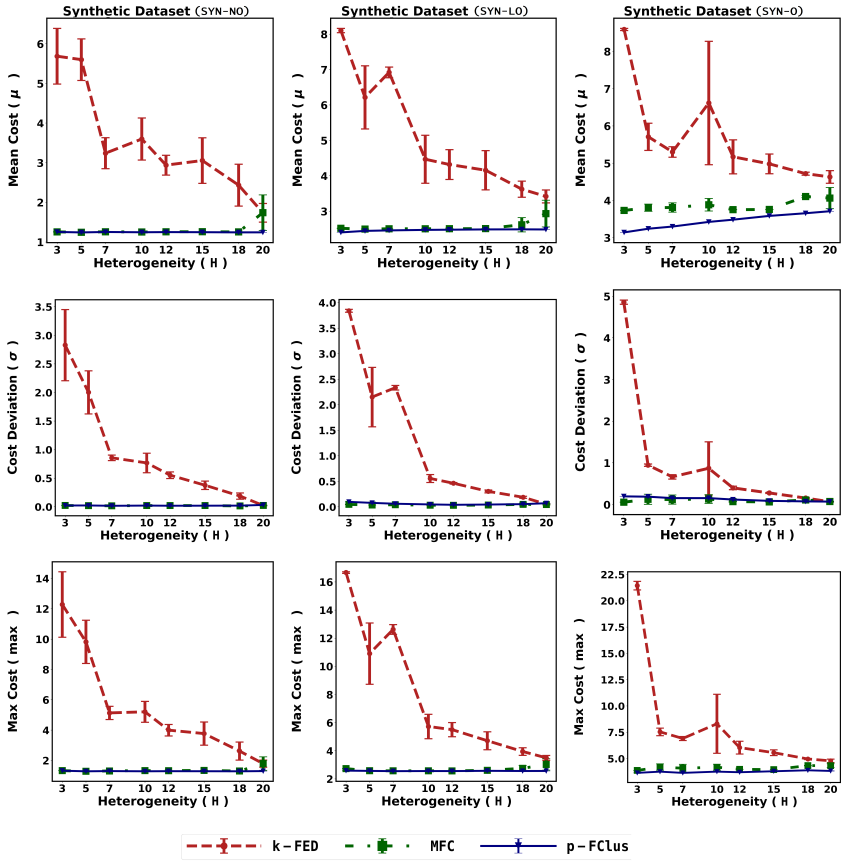


Figure 6.6: The plot shows the variation in evaluation metrics for proposed `p-FClus`, `MFC` and `SOTA` on k-means Objective for varying heterogeneity levels on a `Unequal` data split across 50 clients. Each column represents a specific Synthetic dataset (`Syn`) in sequence: `Syn-NO`, `Syn-LO`, `Syn-O` respectively, and each row represents one metric under evaluation. (Best viewed in color).

6.7.3 Analysis on `Unequal` Data Distribution among Clients on k-means Objective

This subsection now delves into the results of the unequal data distribution setting in the k-means objective ($p = 2$). The results are illustrated in Figure 6.4 to 6.5 for real-world datasets and Figure 6.6 for Synthetic datasets. We first provide a brief overview of the observations per dataset and then summarize the overall results in this setting.

Observations for `Adult`

The $\boldsymbol{\mu}$ is considerably lower for `p-FClus` compared to other methods. However, the performance of `p-FClus` becomes slightly questionable when it comes to the $\boldsymbol{\sigma}$ metric since $\boldsymbol{\sigma}$ is higher for `k-FED` and `MFC` at smaller `H` values but `MFC`'s value drops below `p-FClus` as `H` increases. However, there is high variability in $\boldsymbol{\sigma}$ for `MFC` and `SOTA` methods, whereas the proposed `p-FClus` maintains a reasonable confidence interval. Further, it is least affected by the number of total clients and the heterogeneity level that are unknown in most scenarios. Thus showcasing the adaptability of `p-FClus` seamlessly to many real-world

Dataset	Method	$\boldsymbol{\mu}$ (↓)	$\boldsymbol{\sigma}$ (↓)	**max** (↓)
WISDM	k-FED	$5.71 \times 10^{12} \pm 8.02 \times 10^{10}$	$1.32 \times 10^{13} \pm 1.88 \times 10^{9}$	$7.21 \times 10^{13} \pm 1.32 \times 10^{10}$
	MFC	$3.35 \times 10^{12} \pm 8.32 \times 10^{6}$	$0.25 \times 10^{13} \pm 8.68 \times 10^{6}$	$1.10 \times 10^{13} \pm 8.17 \times 10^{7}$
	p-FClus	$0.56 \times 10^{12} \pm 5.28 \times 10^{5}$	$0.02 \times 10^{13} \pm 8.05 \times 10^{5}$	$0.12 \times 10^{13} \pm 8.33 \times 10^{6}$
FEMNIST	k-FED	2.806 ± 0.0022	0.3518 ± 0.0008	4.4780 ± 0.0130
	MFC	3.158 ± 0.0341	0.3110 ± 0.0186	4.3710 ± 0.0427
	p-FClus	3.103 ± 0.0048	0.3640 ± 0.0035	4.1700 ± 0.0513

Table 6.1: The table summarizes mean and deviation of evaluation metrics for proposed p-FClus, MFC and SOTA on k-means for the Intrinsic datasets. The results are not evaluated for CentClus owing to large main memory requirements (e.g. 8 TB in FEMNIST) and can only be processed using streaming or federated setups.

applications without worrying about the system's heterogeneity level. Also, it should be noted that the slightly higher $\boldsymbol{\sigma}$ is for a significantly reduced mean cost. Thus, when $\boldsymbol{\mu}$ and $\boldsymbol{\sigma}$ are considered together, it demonstrates the efficacy of p-FClus.

Observations for Bank

The trend for $\boldsymbol{\sigma}$ is arbitrary in Figure 6.4 to 6.5; there is no perfect demarcation indicating after how much heterogeneity level the SOTA methods will start performing considerably well on fairness metrics such as $\boldsymbol{\sigma}$ and **max**. In contrast, p-FClus is least affected by heterogeneity level perturbations and the system's number of clients.

Observations for Diabetes

p-FClus has wide gap in $\boldsymbol{\mu}$ and **max** metrics. Further, it has a lower variance in $\boldsymbol{\sigma}$ compared to SOTA, exhibiting the efficacy of the approach.

Observations for Synthetic (Syn)

In unequal data distribution for Syn dataset, it appears like both MFC and p-FClus are quite similar and fairer approaches and only when there is a lot of overlap, i.e., Syn-0, one can see a slight increase in mean cost for MFC, but as seen throughout the section, the results do not follow the similar trend on other datasets, especially real-world datasets. Thus, this helps us lead to the following overall analysis:

Overall Insight: Similar to a balanced (or equal) data split, p-FClus is more likely to be chosen for real-world deployments due to its consistent reliability in terms of the range of cost deviations that different clients may experience. This is because if some clients face high costs, they may lose incentives to stay in the system and could opt to leave. Nonetheless, the performance of p-FClus is also not subject to heterogeneity levels and number of clients. The key factor driving the performance of the proposed method is its personalized approach through fine-tuning steps. Next follows the performance of MFC on cost and deviation compared to k-FED.

Dataset	Method	Metric	Heterogeneity (H)			
			2	5	7	10
100 Clients						
Adult	p-FClus	$\boldsymbol{\mu}(\downarrow)$	1.21 ± 0.00	1.31 ± 0.00	1.31 ± 0.00	1.31 ± 0.00
	CentClus		1.64 ± 0.00			
	p-FClus	$\boldsymbol{\sigma}(\downarrow)$	0.12 ± 0.00	0.06 ± 0.00	0.07 ± 0.00	0.03 ± 0.00
	CentClus		0.03 ± 0.00			
	p-FClus	$\mathbf{max}(\downarrow)$	1.39 ± 0.00	1.45 ± 0.00	1.50 ± 0.00	1.36 ± 0.00
	CentClus		0.60 ± 0.00			
Bank	p-FClus	$\boldsymbol{\mu}(\downarrow)$	1.09 ± 0.00	1.54 ± 0.00	1.61 ± 0.00	1.58 ± 0.00
	CentClus		0.61 ± 0.00			
	p-FClus	$\boldsymbol{\sigma}(\downarrow)$	0.21 ± 0.00	0.12 ± 0.00	0.13 ± 0.00	0.036 ± 0.00
	CentClus		0.14 ± 0.00			
	p-FClus	$\mathbf{max}(\downarrow)$	1.53 ± 0.00	1.82 ± 0.00	2.42 ± 0.00	1.64 ± 0.00
	CentClus		0.94 ± 0.00			
500 Clients						
Adult	p-FClus	$\boldsymbol{\mu}(\downarrow)$	1.08 ± 0.00	1.28 ± 0.00	1.27 ± 0.00	1.29 ± 0.00
	CentClus		1.64 ± 0.00			
	p-FClus	$\boldsymbol{\sigma}(\downarrow)$	0.17 ± 0.00	0.10 ± 0.00	0.11 ± 0.00	0.05 ± 0.00
	CentClus		0.03 ± 0.00			
	p-FClus	$\mathbf{max}(\downarrow)$	1.49 ± 0.00	1.51 ± 0.00	1.45 ± 0.00	1.43 ± 0.00
	CentClus		0.60 ± 0.00			
Bank	p-FClus	$\boldsymbol{\mu}(\downarrow)$	1.16 ± 0.00	1.55 ± 0.00	1.60 ± 0.00	1.58 ± 0.00
	CentClus		0.61 ± 0.00			
	p-FClus	$\boldsymbol{\sigma}(\downarrow)$	0.23 ± 0.00	0.10 ± 0.00	0.07 ± 0.00	0.04 ± 0.00
	CentClus		0.14 ± 0.00			
	p-FClus	$\mathbf{max}(\downarrow)$	1.79 ± 0.00	1.87 ± 0.00	1.76 ± 0.00	1.69 ± 0.00
	CentClus		0.94 ± 0.00			

Table 6.2: The table summarizes mean and deviation of evaluation metrics for proposed p-FClus, MFC and CentClus on k-medoids for varying heterogeneity levels on Balanced data split across 100 and 1000 clients.

6.7.4 Analysis on Intrinsic Federated Datasets on k-means Objective

This subsection delves into datasets with a pre-captured level of heterogeneity (H). This experiment directly compares the SOTA with p-FClus on performance metrics.

WISDM

The results for the WISDM dataset are summarized in Table. 6.1. It can be observed that MFC and p-FClus against k-FED have a wide gap in $\boldsymbol{\sigma}$ and **max**, owing to achieving a fair solution as a byproduct or through fine-tuning steps, respectively. The performance of $\boldsymbol{\mu}$ is also significantly reduced by an order of 10^5 times, indicating that our p-FClus is the best available fair federated solution.

`FEMNIST`

The results for the `FEMNIST` dataset are summarized in Table. 6.1. `k-FED` is slightly better on cost. On the other hand, `p-FClus` and `MFC` have comparable performance on cost. The performance of all methods is quite similar, possibly due to the nature of the dataset. However, `p-FClus` is considerably close in $\boldsymbol{\mu}, \boldsymbol{\sigma}$, but it has a lower **max** cost a client has to suffer, thus overtaking `SOTA` methods. The performance of `MFC` follows next after `p-FClus` on maximum cost.

Dataset	Method	Metric	Heterogeneity (`H`)			
			2	5	7	10
100 Clients						
`Adult`	`p-FClus`	$\boldsymbol{\mu}(\downarrow)$	1.17 ± 0.00	1.31 ± 0.00	1.32 ± 0.00	1.29 ± 0.00
	`CentClus`		1.64 ± 0.00			
	`p-FClus`	$\boldsymbol{\sigma}(\downarrow)$	0.11 ± 0.00	0.11 ± 0.00	0.07 ± 0.00	0.16 ± 0.00
	`CentClus`		0.03 ± 0.00			
	`p-FClus`	$\mathbf{max}(\downarrow)$	1.46 ± 0.00	1.48 ± 0.00	1.47 ± 0.00	1.65 ± 0.00
	`CentClus`		0.60 ± 0.00			
`Bank`	`p-FClus`	$\boldsymbol{\mu}(\downarrow)$	1.21 ± 0.00	1.50 ± 0.00	1.63 ± 0.00	1.71 ± 0.00
	`CentClus`		0.61 ± 0.00			
	`p-FClus`	$\boldsymbol{\sigma}(\downarrow)$	0.14 ± 0.00	0.11 ± 0.00	0.06 ± 0.00	0.10 ± 0.00
	`CentClus`		0.14 ± 0.00			
	`p-FClus`	$\mathbf{max}(\downarrow)$	1.55 ± 0.00	1.80 ± 0.00	1.75 ± 0.00	2.04 ± 0.00
	`CentClus`		0.94 ± 0.00			
500 Clients						
`Adult`	`p-FClus`	$\boldsymbol{\mu}(\downarrow)$	1.17 ± 0.00	1.30 ± 0.00	1.29 ± 0.00	1.27 ± 0.00
	`CentClus`		1.64 ± 0.00			
	`p-FClus`	$\boldsymbol{\sigma}(\downarrow)$	0.11 ± 0.00	0.09 ± 0.00	0.07 ± 0.00	0.14 ± 0.00
	`CentClus`		0.03 ± 0.00			
	`p-FClus`	$\mathbf{max}(\downarrow)$	1.38 ± 0.00	1.50 ± 0.00	1.48 ± 0.00	1.65 ± 0.00
	`CentClus`		0.60 ± 0.00			
`Bank`	`p-FClus`	$\boldsymbol{\mu}$ $(\downarrow)$	1.17 ± 0.00	1.53 ± 0.00	1.63 ± 0.00	1.54 ± 0.00
	`CentClus`		0.61 ± 0.00			
	`p-FClus`	$\boldsymbol{\sigma}$ $(\downarrow)$	0.18 ± 0.00	0.12 ± 0.00	0.08 ± 0.00	0.19 ± 0.00
	`CentClus`		0.14 ± 0.00			
	`p-FClus`	$\mathbf{max}(\downarrow)$	1.75 ± 0.00	1.90 ± 0.00	1.79 ± 0.00	2.11 ± 0.00
	`CentClus`		0.94 ± 0.00			

Table 6.3: The table summarizes mean and deviation of evaluation metrics for proposed `p-FClus`, `MFC` and `CentClus` on k-medoids for varying heterogeneity levels on `Unequal` data split across 100 and 500 clients.

6.7.5 Analysis on different Dataset for k-mediod Objective

In k-mediod, we limit the comparison only to `CentClus` because other baselines are not intrinsically designed to handle objectives except k-means. The results for `Balanced` data split for real-world non-federated datasets are reported in Table 6.2 for $100, 1000$ clients. We can clearly observe that the mean per point cost ($\boldsymbol{\mu}$), and correspondingly max cost (**max**) for `p-FClus` is quite close to centralized clustering, showcasing that `p-FClus` efficiently captures the centers in a federated setting. Furthermore, it not only showcases its efficacy in cost but also in the fairness metric, i.e., $\boldsymbol{\sigma}$, which is also considerably low across clients, thus resulting in a fair aka personalized clustering.

Dataset	Method	Metric	Value		Dataset	Value
WISDM	**p-FClus**	$\boldsymbol{\mu}(\downarrow)$	$1.08 \times 10^{13} \pm 0.00$		FEMNIST	6.87 ± 0.00
	CentClus		(refer caption)			2.91 ± 0.00
	p-FClus	$\boldsymbol{\sigma}(\downarrow)$	$2.64 \times 10^{13} \pm 0.00$			0.96 ± 0.00
	CentClus		(refer caption)			0.65 ± 0.00
	p-FClus	**max**$(\downarrow)$	$1.30 \times 10^{14} \pm 0.00$			9.69 ± 0.00
	CentClus		(refer caption)			4.32 ± 0.00

Table 6.4: The table summarizes mean and deviation of evaluation metrics for proposed `p-FClus`, `MFC` and `CentClus` on k-medoids for Intrinsic datasets. The `WISDM` dataset is not evaluated on `CentClus` due to 8 terabytes of main memory requirements.

The results for `Unequal` data split are reported in Table 6.3 for 100 and 500 clients. In this setting also, the observations are similar to the previous setting, showcasing the benefit of `p-FClus` in both scenarios. The results for intrinsic federated datasets, namely `WISDM` and `FEMNIST`, are reported in Table 6.4. We do not report the `CentClus` results for the `WISDM` dataset as the main memory requirement for such a large dataset is nearly 8TB and thus can only be processed in streaming or federated (distributed) settings.

6.8 Conclusion and Future Directions

In this chapter, we focus on solving the problem of handling unlabelled data in a federated setting. We propose `MFC` that, unlike prior methods does not rely on the data distribution across clients and under well separability assumptions (similar to [109, 41]), we have theoretical bounds on the gap between local and global centers obtained using `MFC`. We further propose a first-of-its-kind personalized data clustering algorithm, `p-FClus` which operates in three sub-phases within a single round of to-and-fro communication between the server and clients. Furthermore, through rigorous experimental analysis, we observe that `p-FClus`'s performance does not suffer across varying levels of heterogeneity and clients (dataset sizes), showcasing its data distribution independence. Additionally, it achieves a lower mean per-point objective cost in most scenarios compared to `SOTA` methods while ensuring small deviations in cost across clients (fairness). Even the maximum cost any client incurs using `p-FClus` is significantly lower than `SOTA` methods in almost all

settings, enabling clients to have personalized models and incentivizing them to continue contributing to the federated setup. Moreover, the method is reliable and applicable to any finite p-norm objectives, including k-means and k-medoids. An immediate future direction involves studying a robust data clustering method in the presence of malicious clients [339] and noisy data [319]. Other interesting directions include investigating scenarios where clients might strategically report their features, thus hampering the quality of generated local centers [340]. Since in federated learning, clients can join and leave the system, therefore looking into the direction of unlearning information from the global model in clustering can be another promising direction [341].

Chapter 7

Mitigating Popularity Bias in Recommender Systems

Abstract

Recommender systems are unsupervised machine learning methods that are deployed heavily by many online platforms for better user engagement and providing recommendations. Despite being so popular, several works have shown the existence of popularity bias due to the non-random nature of missing data. Popularity bias leads to the recommendation of only a few popular items (majority), causing starvation of many non-popular items (minority). This chapter considers an easy-to-understand metric to evaluate the popularity bias as the difference between mean squared error on popular and non-popular items. Then, we propose EqBal-RS, a novel re-weighting technique that updates the weights of popular and non-popular items. Re-weighting ensures that both item sets are equally balanced during training using a trade-off function between overall loss and popularity bias. This is analogous to balancing the trade-off between cost and group fairness in clustering literature. Our experiments on real-world datasets show that EqBal-RS outperforms the existing state-of-the-art algorithms in terms of accuracy, quality, and fairness. EqBal-RS works well on the proposed and existing popularity bias metrics and has significantly reduced runtime. The code is publicly available at https://github.com/eqbalrs/EqBalRS

7.1 Introduction

Online platforms, including books [342], movies, and music streaming platforms (Netflix, Spotify) [343, 344, 345, 346, 347], e-commerce websites (Amazon), Third party libraries [348] and even social media platforms (Instagram), face the choice overload problem [349, 345]. Recommender Systems are useful unsupervised learning tools that efficiently solve this problem by refining the information according to the users' choices. The central goal of recommender systems is to recommend items that the user might like by predicting the pertinence of items with which the user has never interacted based on the user's past behavior. Past studies have shown that decision support systems based on previous

This chapter has been published as a full paper in the Journal of Intelligent Information Systems (JIIS)[285].

user behavior can unconsciously inherit existing human biases and introduce new ones [350, 351, 352, 353, 354, 355]. This raises several fairness issues in recommender systems, primarily on the user [356, 357, 77, 75, 358, 73, 359] and item sides [360, 361, 71]. This chapter aims to mitigate the popularity bias [362, 363] in recommender systems at the item side.

Popularity bias occurs when popular items (i.e., items with high rating frequency) are recommended more often to the users than other items, even if the user has a reasonable interest in the latter. The primary reason behind this is that popular items have better representation in the training data used by optimization procedures. While aiming to reduce average loss, these procedures might lead to a biased model [362]. Thus, this provokes the recommendation of similar items to most users, even if they are apathetic, and new (non-popular) items might starve for desired visibility [134, 364]. Over-reliance on recommending popular items could also negatively affect businesses. In fact, several U.S. states recently filed a lawsuit against Google U.S. for advertising popular items and giving lesser visibility to newer items [29]. This practice can create an exclusive market position for certain items, posing challenges for firms and stifling innovation in product development. Note that handling popularity bias is quite analogous to group fairness in clustering. The analogy can be apparent by considering the majority group as popular items and the minority group as non-popular items. The goal is to balance each group's representation in the recommendation list. Note that in recommender systems, one can not arbitrarily enforce the representation of minorities; one must consider user preferences and history.

Many past works [123, 121] have explored and ameliorated popularity bias by improving the model's overall accuracy or the diversity among non-popular items. Accuracy is innately biased for popular items as these items are rated more frequently. A highly accurate model might suffer a heavy loss on non-popular items. On the contrary, a method that naively improves diversity might generate an overall poorly accurate recommendation model. Rather, an unbiased recommender system should perform well on popular and non-popular items, thus resulting in a fair and accurate recommender system. To this, we first propose a novel metric Popularity Parity that enables a scalable algorithm. Next, we propose EqBal-RS, a Matrix Factorization (MF) based algorithm that balances the losses on popular and non-popular items. Existing re-weighting techniques [27, 8] use propensity scores that require careful investigation among different available score criteria as exposure mechanism is rarely known, making them dataset-dependent. Further, these approaches require heavy pre-training to compute the weights (scores) accurately, and the approach exhibits high variance. Our EqBal-RS does not require such pre-training as it inherently learns the weights while training the overall model. To summarize, we list down our **contributions** below:

- Our novel metric, Popularity Parity, measures popularity bias as the difference in the Mean Squared Error (MSE) on the popular and non-popular items.

- We propose EqBal-RS, a novel technique that solves the optimization problem of reducing overall loss with a penalty on popularity bias. It does not require any heavy pre-training.

- Through extensive experiments on real-world datasets (MovieLens, Yahoo, and Amazon GiftCards), we show that EqBal-RS outperforms existing approaches on recommendation accuracy, quality, and fairness. It works exceptionally well on Popularity Parity while having comparable performance on existing metrics like Average Rating Popularity (ARP) and Normalized Discounted Cumulative Gain (NDCG). Further, it does not compromise on the diversity of items.

Roadmap: The remainder of the chapter is organized as follows: Section 7.2 reviews the existing literature. Section 7.3 provides an overview of the different notations and definitions used throughout the chapter. These will be helpful for understanding the proposed methodology in Section 7.4. We validate the efficacy of EqBal-RS on the proposed metric Popularity Parity and existing metrics against state-of-the-art (`SOTA`) methods in Section 7.5. Finally, Section 7.6 concludes the work.

7.2 Related Work

The work closest to our approach is to mitigate the popularity bias by using Inverse Propensity Scores (IPS) [8, 360]. The score helps in generating a pseudo missing completely at random dataset by weighting all the observed ratings. Although IPS loss is proven to be an unbiased estimator, these methods majorly suffer from two problems. First, the IPS estimator might become biased if the propensity estimation model is not appropriately stated. Second, IPS estimators suffer from high variance as the inverse of the propensities might be substantial. To overcome these challenges, [27] proposed an asymmetric tri-training technique. It involves three rating predictors, two of which create a pseudo-rating dataset, and the third trains the model on these pseudo-ratings. The main limitation is that it becomes impossible to estimate the ratings of all items accurately as the dataset size reduces after applying the technique. Thus, there is a need for an effective strategy to tackle popularity bias in recommender systems.

7.3 Preliminaries

Consider the data with $\mathcal{U}$ denoting the set of users and $\mathcal{I}$ be the set of items. Let $R = \mathcal{U} \times \mathcal{I}$ be a rating matrix where each entry $R_{u,i}$ corresponds to the true rating of item i by the user on a scale of 1 (lowest) to 5 (highest). All non-interacted user-item (i, j) pairs have a value of $R_{u,i} = 0$. Let $\mathcal{I}_{\mathcal{P}}$ and $\mathcal{I}_{\mathcal{NP}} = \mathcal{I} \setminus \mathcal{I}_{\mathcal{P}}$ denote the set popular items and non-popular items respectively. Inspired by Abdollahpouri et al. [118], we use a threshold mechanism to generate $\mathcal{I}_{\mathcal{P}}$ and $\mathcal{I}_{\mathcal{NP}}$. Motivated by Pareto principle [119, 120], we set the threshold as top 20% items in terms of rating frequency as popular and remaining as

non-popular (long tail) items. The prediction matrix is given by P with each entry $P_{u,i}$ as the predicted rating for user u and item i. The goal of an ideal recommendation algorithm is to reduce the following loss function:

$$L_{\text{ideal}}(R, P) = \frac{1}{|\mathcal{U}||\mathcal{I}|} \sum_{u \in \mathcal{U}} \sum_{i \in \mathcal{I}} \delta(R_{u,i}, P_{u,i}) \tag{7.1}$$

where the error function δ could be the mean squared error (MSE) or mean absolute error (MAE). Since the true rating $R_{u,i}$ is not available for all possible user-item interactions, one tends to minimize the loss on the observed set of user-item interactions given by:

$$L_{\text{obs}}(R, P) = \frac{\sum_{u \in \mathcal{U}, i \in \mathcal{I}} \mathbb{I}(R_{u,i} \neq 0)\ \delta(R_{u,i}, P_{u,i})}{\sum_{u \in \mathcal{U}, i \in \mathcal{I}} \mathbb{I}(R_{u,i} \neq 0)} \tag{7.2}$$

The algorithms that aim solely to minimize L_{obs} can pick popularity bias from the dataset. It is because inherently popular items are rated more frequently and are available more in the dataset. We now formalize a novel popularity metric Popularity Parity (PP), which quantifies popularity bias as the difference between losses on non-popular and popular items. The intuition is that by minimizing both these losses while ensuring overall loss minimization, one can expect fair visibility of all items in the final recommendation list. Let,

$$L_{\mathcal{NP}}(R, P) = \frac{\sum_{(u,i):\ i \in \mathcal{I}_{\mathcal{NP}}} \mathbb{I}(R_{u,i} \neq 0)\ \delta(R_{u,i}, P_{u,i})}{\sum_{(u,i):\ i \in \mathcal{I}_{\mathcal{NP}}} \mathbb{I}(R_{u,i} \neq 0)} \tag{7.3}$$

and,

$$L_{\mathcal{P}}(R, P) = \frac{\sum_{(u,i):\ i \in \mathcal{I}_{\mathcal{P}}} \mathbb{I}(R_{u,i} \neq 0)\ \delta(R_{u,i}, P_{u,i})}{\sum_{(u,i):\ i \in \mathcal{I}_{\mathcal{P}}} \mathbb{I}(R_{u,i} \neq 0)} \tag{7.4}$$

define the loss on non-popular items and popular items, respectively. Then the Popularity Parity is given as:

$$PP(R, P) = L_{\mathcal{NP}}(R, P) - L_{\mathcal{P}}(R, P) \tag{7.5}$$

We now propose a fair MF-based approach–**EqBal-RS** with significantly reduced runtime while having comparable performance on loss (accuracy).

7.4 Proposed Algorithm: EqBal-RS

Equally Balancing Recommender System (**EqBal-RS**) presented in Algorithm 17 is a collaborative filtering-based technique. It assigns a weight to every item during learning of user and item embeddings and trains the model towards equalizing the balance between loss on popular and non-popular items. Past work on weighting techniques uses inverse propensity scores. However, methods can suffer heavily from popularity bias and losses if the scores are not tuned properly. EqBal-RS automatically updates the weights computed using an objective function that minimizes the overall weighted loss and Popularity Parity. For a given weight vector $\mathbf{w} = \{w_i\}_{i \in \mathcal{I}}$, the combined loss function

is given by:

$$Z(\mathbf{w}) = \sum_{u \in \mathcal{U}} \sum_{i \in \mathcal{I}} w_i (R_{i,u} - P_{i,u})^2 + \Upsilon \left(PP_{\mathbf{w}}(R, P) \right)^2 \tag{7.6}$$

Here, $PP_{\mathbf{w}}(R, P)$ represents the weighted Popularity Parity with the loss for each item i being weighted by w_i. For the sake of simplicity, we omit the indicator function, which observes the presence of a true rating $R_{u,i} > 0$. To avoid the model getting biased towards popular items, we take a square of weighted popularity bias. Further, let

$$l_i(\mathbf{w}) = \sum_{u \in \mathcal{U}} w_i (R_{i,u} - P_{i,u})^2$$

represent the weighted loss on item i. Let $C_{\mathcal{NP}} = \sum_{u \in \mathcal{U}, i \in \mathcal{I}_{\mathcal{NP}}} \mathbb{I}(R_{u,i} \neq 0)$ and $C_{\mathcal{P}} = \sum_{u \in \mathcal{U}, i \in \mathcal{I}_{\mathcal{P}}} \mathbb{I}(R_{u,i} \neq 0)$ denote the number of ratings obtained for non-popular and popular items respectively. Then weighted Popularity Parity is given as:

$$PP_{\mathbf{w}}(R, P) = \frac{\sum_{i \in \mathcal{I}_{\mathcal{NP}}} l_i(\mathbf{w})}{C_{\mathcal{NP}}} - \frac{\sum_{i \in \mathcal{I}_{\mathcal{P}}} l_i(\mathbf{w})}{C_{\mathcal{P}}} \tag{7.7}$$

On substitution,

$$Z(\mathbf{w}) = \sum_{i \in \mathcal{I}} l_i(\mathbf{w}) + \Upsilon \left(\frac{\sum_{i \in \mathcal{I}_{\mathcal{NP}}} l_i(\mathbf{w})}{C_{\mathcal{NP}}} - \frac{\sum_{i \in \mathcal{I}_{\mathcal{P}}} l_i(\mathbf{w})}{C_{\mathcal{P}}} \right)^2 \tag{7.8}$$

The parameter Υ is the trade-off between Popularity Parity and overall squared loss. The intuition behind optimizing the weighted loss function is: If the weights of non-popular items are lower than the popular item, i.e., the non-popular items are under-represented. In such cases, though the model will give a good overall accuracy, it will suffer badly on the weighted Popularity Parity. Similarly, if popular items are under-represented, the model will still suffer badly from the weighted popularity metric. Thus, the given loss function will try to push for equal representation of both popular and non-popular items by updating weights to point towards the direction of minima for loss function $Z(\mathbf{w})$.

The well-known gradient descent technique can compute the weight update equations. The idea is to break the first term in Z separately for popular and non-popular items. Then in the case of non-popular item i, we get,

$$\left(\frac{\partial Z}{\partial w_i} \right)_{i \in \mathcal{I}_{\mathcal{NP}}} = \sum_{u \in \mathcal{U}} (R_{u,i} - P_{u,i})^2 \left(1 + \frac{2 \ \Upsilon \ \mathrm{PP}_{\mathbf{w}}(R, P)}{C_{\mathcal{NP}}} \right) \tag{7.9}$$

For popular item i, one can easily find,

$$\left(\frac{\partial Z}{\partial w_i} \right)_{i \in \mathcal{I}_{\mathcal{P}}} = \sum_{u \in \mathcal{U}} (R_{u,i} - P_{u,i})^2 \left(1 - \frac{2 \ \Upsilon \ \mathrm{PP}_{\mathbf{w}}(R, P)}{C_{\mathcal{P}}} \right) \tag{7.10}$$

Let $\Delta_i = \left(\frac{\partial Z}{\partial w_i} \right)_{i \in \mathcal{I}_{\mathcal{NP}}}, \forall i \in \mathcal{I}_{\mathcal{NP}}$ and $\Delta_i = \left(\frac{\partial Z}{\partial w_i} \right)_{i \in \mathcal{I}_{\mathcal{P}}}, \forall i \in \mathcal{I}_{\mathcal{P}}$ denote the derivative of item i. We will use these to update the weights of items.

Algorithm 17: EQBAL-RS

Input: given items (popular and non-popular) $\mathcal{I}$=$\mathcal{I}_{\mathcal{P}} \cup \mathcal{I}_{\mathcal{NP}}$, users $\mathcal{U}$, rating matrix R, time-steps T, trade-off Υ, learning rate η, epochs E, wt-decay Λ, latent factors κ

Output: learned user and item embedding ξ_t, ψ_t respectively

1 Initialize item weights array $\mathbf{w}$ with $w_i = \frac{1}{|\mathcal{I}|} \forall i \in \mathcal{I}$

2 Randomly initialize $\xi_0 \in \mathbb{R}^{|\mathcal{U}| \times \kappa}$ and $\psi_0 \in \mathbb{R}^{\kappa \times |\mathcal{I}|}$.

3 $m_{u_0}, v_{u_0}, m_{i_0}, v_{i_0} \leftarrow 0$ (Initialize 1^{st}, 2^{nd} moment vectors for user and items.)

4 **for** $t \leftarrow 1$ **to** T **do**

5 $\quad \xi_t, \psi_t, m_{u_t}, v_{u_t}, m_{i_t}, v_{i_t} =$ **MFAdam**$(\xi_{t-1}, \psi_{t-1}, m_{u_{t-1}}, v_{u_{t-1}}, m_{i_{t-1}}, v_{i_{t-1}}, \eta, \mathbf{w}, E, R)$

6 $\quad P^t = \xi_t \times \psi_t$

7 $\quad$ Calculate weighted POPULARITY PARITY $PP^t_{\mathbf{w}}(R, P^t)$

8 $\quad$ Calculate gradient $\Delta_i \ \forall i \in I$ using Equations 7.9, 7.10 and $PP^t_{\mathbf{w}}$

9 $\quad$ Update item weights array $\mathbf{w}$ using $w_i = w_i - \Lambda \times \Delta_i, \ \forall i \in \mathcal{I}$

10 **end**

11 **return** ξ_t, ψ_t

We now describe EQBAL-RS given in Algorithm 17. Inputs given to the algorithm involve a set of popular ($\mathcal{I}_{\mathcal{P}}$) and non-popular ($\mathcal{I}_{\mathcal{NP}}$) items, learning rate η, and weight decay parameter Λ. We use κ to denote the number of latent factors. Line 1 initialize item weights (equal), user embedding (ξ) of size $|\mathcal{U}| \times \kappa$ (random), item embedding (ψ) of size $\kappa \times |\mathcal{I}|$ (random), and adam optimization-related moment vectors (set to 0). The dot product of user and item embedding gives rise to the prediction matrix P. The moment vectors, user, and item embedding will be learned in-processing continually over timesteps. The model training starts from line 2 for T timesteps. The embeddings learned up to the current timestep are passed to procedure MFAdam (presented in Algorithm 18 and described later). After the procedure completes (say E epochs), it returns the learned embeddings and moment vectors. These embeddings compute prediction matrix P^t at timestep t, which helps evaluate weighted popularity bias and weight updates using Equations 7.9, 7.10. At the end of T time-steps, the algorithm returns the user and item embedding independent of weights. Thus, we only need the two embeddings for post-training prediction and do not require any weights or propensity scores.

MFAdam described in Algorithm 18 clubs the ideas from traditional MF and Adam optimization [365]. It essentially uses adam optimization to learn embedding while minimizing weighted squared error $\sum_{i \in \mathcal{I}} l_i(\mathbf{w})$. The algorithm also takes item weights and moment vectors passed via EQBAL-RS. These improvise the embeddings learned till previous timesteps. Once all computations are over, the procedure returns updated embeddings and moment vectors.

We use adam optimizer as it can handle sparse data well, and default parameter configurations are adaptable to many problems [365]. Thus, it widens the usability of EQBAL-RS to different applications and datasets. Further, the base chapter shows that Adam works much faster than stochastic gradient descent. It will help reach minima

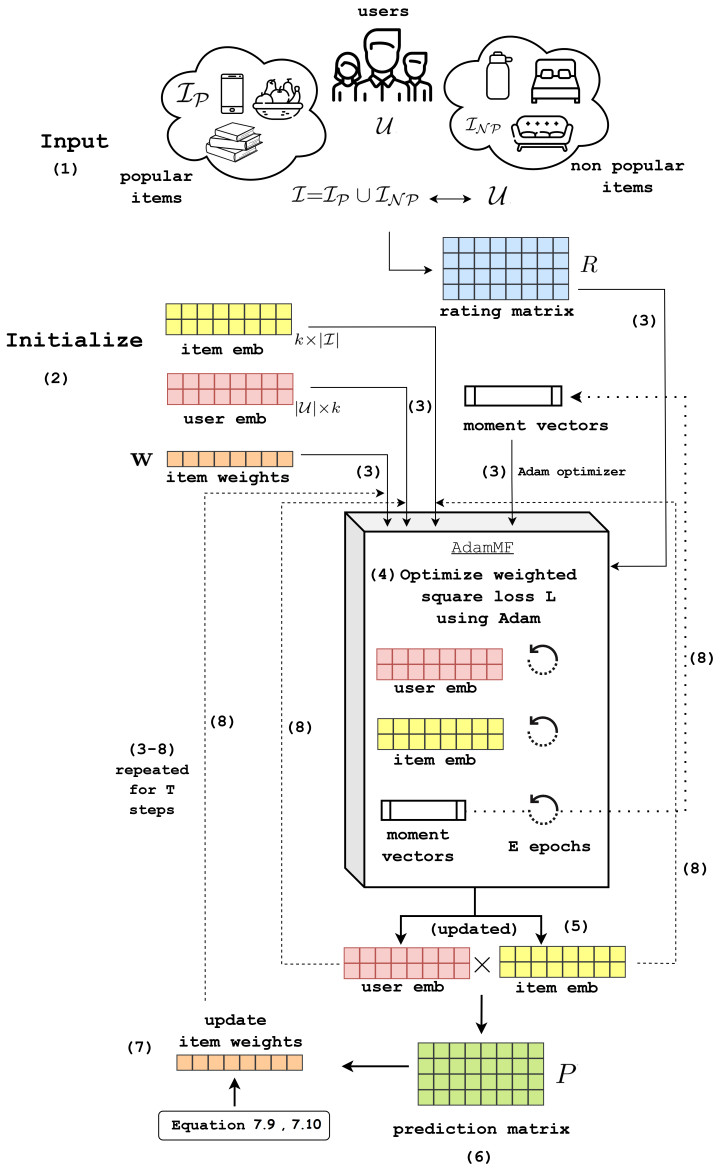


Figure 7.1: Flowchart for working of proposed EqBal-RS. (Best viewed in color)

quickly in each timestep before the next weight update. Figure 7.1 visually represents the entire algorithmic workflow.

7.5 Experimental Result and Discussion

We will now evaluate the performance of EqBal-RS against different state-of-the-art (SOTA) techniques for recommendation systems on three *bench-marking datasets* of various sizes, (i) MovieLens[1] is a movie rating dataset with 1 Million ratings given to 3706 movies by 6040 users, (ii) Yahoo[2] provides 365,000 ratings to 1000 music items by 15,400 users, (iii) Amazon GiftCards[3] is an amazon dataset with 147,000 ratings given to 1548 gift

[1]https://grouplens.org/datasets/movielens
[2]https://webscope.sandbox.yahoo.com
[3]https://nijianmo.github.io/amazon

Algorithm 18: MFAdam: Matrix Factorization with Adam Optimization

Input: embeddings ξ_{t-1}, ψ_{t-1} & moments $m_{u_{t-1}}$,$v_{u_{t-1}}$,$m_{i_{t-1}}$, $v_{i_{t-1}}$, R, learning rate η, epochs E, weights $\mathbf{w} = \{w_i\}_{i \in \mathcal{I}}$, and Adam's hyper-parameters β_1, β_2 and ϵ (default settings: $\eta = 0.001$, β_1=0.9, β_2=0.999, ϵ=10^{-8}).

Output: learned embeddings ξ_t, ψ_t, & moments m_{u_t},v_{u_t} ,m_{i_t} , v_{i_t}

$ep \leftarrow 0$ (Current epoch number)
while $ep \leq E$ **do**
  Now let overall weighted loss function be L
  $L = \sum_{i \in \mathcal{I}} l_i(\mathbf{w})\ \forall i \in \mathcal{I}$.
  Use Adam optimizer on L (refer [365])
  ξ_t, ψ_t, m_{u_t},v_{u_t} ,m_{i_t}, v_{i_t}=**Adam**(ξ_{t-1}, ψ_{t-1}, $m_{u_{t-1}}$,$v_{u_{t-1}}$,$m_{i_{t-1}}$, $v_{i_{t-1}}$, η)
  $ep \leftarrow ep + 1$
end
return ξ_t, ψ_t, m_{u_t}, v_{u_t}, m_{i_t}, v_{i_t}

cards by 128,874 users. We split the dataset into train & test sets (80 : 20) [121, 366]. We compare EqBal-RS against the following **baselines and SOTA** that are MF-based approaches and work with explicit feedback:

- **Matrix Factorization (MF)**: Basic collaborative filtering technique [367].
- **MF with Regularisation (MFR)**: Regularized MF to avoid over-fitting by penalizing the magnitude of user and item vectors [9].
- **MFIPS**: Matrix factorization (MF) with inverse propensity score. We use Naive Bayes as a propensity score estimator in our experiments. The choice is based on findings in the original work that it performs better on given datasets [8].
- **MFIPS-AT**: Improvised MFIPS with naive bayes where MFIPS is used thrice as rating predictor [27]. Note that MFIPS-AT requires pre-training steps due to the involvement of pseudo-labeling.

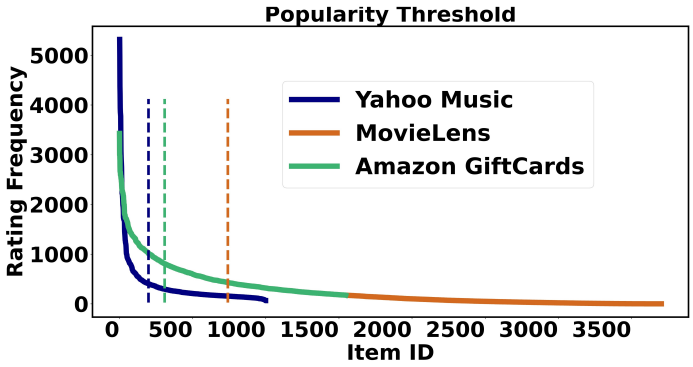


Figure 7.2: Rating frequency and popularity threshold in datasets. (Best viewed in color)

Experimental Setup: All the experiments are executed on an Intel *i*7 CPU and 32GB RAM. We use the optuna framework[4] for hyper-tuning the parameters in all

[4]https://optuna.org

Algorithm	Hyper-parameter	Range Explored	Final Tuned Value		
			MovieLens	**Yahoo Music**	**Amazon**
EqBal-RS	learning Rate (η)	$[10^{-5}, 4]$	4.750	3.785	3.950
	weight-decay (Λ)	$[0, 1)$	1.0×10^{-8}	1.1×10^{-8}	1.01×10^{-8}
	trade-off parameter (Υ)	–	0.010	0.010	0.010
Baselines					
MF	learning Rate (η)	$[10^{-5}, 4]$	0.002300	0.000823	0.008570
MF + Reg	learning Rate (η)	$[10^{-5}, 4]$	0.000128	0.000925	0.007200
	regularisation parameter	$[0, 2)$	0.040000	0.007240	0.004300
MFIPS	regularisation parameter (λ)	$[10^{-6}, 1]$	2.28×10^{-6}	5.2×10^{-5}	0.000009167
	learning rate (η)	$[10^{-8}, 1]$	0.041800	0.31110	0.1.19742
MFIPS-AT	regularisation parameter (λ)	$[10^{-6}, 1]$	5.2099×10^{-5}	5.2×10^{-5}	5.1×10^{-5}
	training parameter (ϵ)	$[10^{-8}, 1]$	0.0047924	0.004792	0.004792
	learning rate (η)	$[10^{-8}, 1]$	0.037870	0.010499	0.030741

Table 7.1: Hyper-parameters for EqBal-RS and baselines tuned using optuna.

the algorithms (see Table 7.1 for details). The code is publicly available as a GitHub repository[5]. We report the mean and standard deviation over ten independent runs. Following the pareto principle in [119, 122, 368], we generate the top 20% of the items with the highest rating count as the popular items and the rest 80% as the non-popular items (see Figure 7.2). The vertical cut represents the popularity threshold.

Evaluation Metrics: The metrics used for comparison include mean square error (MSE) (consistent with literature [8, 27]) on the complete, popular, and non-popular set of items. Along with these three metrics, we also report the results for the absolute value of Popularity Parity, NDCG, and ARP [122]. ARP is quite common in literature but is best suited for learning-to-rank recommender systems in a dynamic setting. However, we intend to tackle algorithmic bias in a static setting. Mathematically:

$$ARP = \frac{1}{|\mathcal{U}|} \sum_{u \in \mathcal{U}} \sum_{i \in L_u} \frac{\omega_i}{|L_u|}$$

where L_u is the recommendation list of user u and ω_i is ratio of number of ratings for item i to total number of items. We use the top 10 recommendation item list for our experimental comparison of ARP. We show that EqBal-RS succeeds on ARP, which helps us claim that reducing Popularity Parity is in line with previous literature [357]. Similarly, NDCG is a widely used standard measure in search ranking evaluations [369] and implicit settings [121, 370]. It is normalized to have a maximum value of 1.0. We find the cumulative NDCG score for each user's top 10 items. A good recommender system should have low losses, popularity bias, ARP, and high NDCG scores.

[5]https://github.com/eqbalrs/EqBalRS

Datasets	Algorithms	Overall MSE(↓)	MSE on $\mathcal{I}_{\mathcal{P}}$(↓)	MSE on $\mathcal{I}_{\mathcal{NP}}$(↓)	PP(↓)	ARP@10(↓)	NDCG@10(↑)	NonPop@10	NonPop@100	t-test(↓)
MovieLens	MF	0.7887 ± 0.0004	0.7737 ± 0.0007	0.7937 ± 0.0004	0.0199 ± 0.0004	0.0684 ± 0.0041	0.0449 ± 0.0045	7.6312 ± 1.3800	75.8358 ± 4.3349	-0.0026 ± 0.0008
	MF+Reg	0.8211 ± 0.0004	0.7974 ± 0.0004	0.8161 ± 0.0004	0.0189 ± 0.0000	0.0399 ± 0.0111	0.00002 ± 0.0005	8.6240 ± 0.8330	83.2096 ± 0.8734	-0.0107 ± 0.0003
	MFIPS	0.6091 ± 0.0154	0.5752 ± 0.0134	0.6111 ± 0.0155	0.0426 ± 0.0043	0.1681 ± 0.0740	0.0049 ± 0.4387	5.1612 ± 1.0170	40.8421 ± 4.9493	0.0011 ± 0.0010
	MFIPS-AT	0.9114 ± 0.0016	0.8759 ± 0.0022	0.9776 ± 0.0021	0.1885 ± 0.0059	0.1266 ± 0.0040	0.0003 ± 0.0012	9.4033 ± 0.5573	78.5129 ± 6.9933	-0.0201 ± 0.0010
	EqBal-RS	0.8358 ± 0.0008	0.8390 ± 0.0011	0.8299 ± 0.0011	0.0091 ± 0.0015	0.0355 ± 0.0078	0.0077 ± 0.0028	8.7360 ± 0.1990	67.5133 ± 0.7099	-0.0046 ± 0.0008
Yahoo	MF	0.5874 ± 0.0054	0.5501 ± 0.0080	0.6333 ± 0.0053	0.0834 ± 0.0089	0.9580 ± 0.012	0.0414 ± 0.0007	2.8770 ± 1.6110	56.1244 ± 3.9817	0.0940 ± 0.0054
	MF + Reg	0.5596 ± 0.0052	0.5063 ± 0.0076	0.6262 ± 0.0054	0.1197 ± 0.0086	0.8442 ± 0.0100	0.0398 ± 0.0006	1.6638 ± 0.0933	56.8050 ± 4.0891	0.0757 ± 0.0051
	MFIPS	0.5479 ± 0.0316	0.4922 ± 0.0432	0.6187 ± 0.0180	0.1391 ± 0.0241	1.2098 ± 0.0425	0.1880 ± 0.0536	1.0656 ± 1.4369	45.3498 ± 10.0136	-0.0024 ± 0.0005
	MFIPS-AT	1.015 ± 0.0145	0.9573 ± 0.0120	1.0890 ± 0.0184	0.2693 ± 0.0212	1.4500 ± 0.0880	0.3112 ± 0.0282	1.2451 ± 1.2189	43.8848 ± 7.1070	-0.0112 ± 0.0005
	EqBal-RS	0.6457 ± 0.0058	0.6104 ± 0.0091	0.6912 ± 0.0034	0.0808 ± 0.0084	0.7851 ± 0.0248	0.0356 ± 0.0016	5.3979 ± 0.1212	63.4142 ± 0.6803	0.0498 ± 0.0076
Amazon	MF	0.0046 ± 0.0007	0.0030 ± 0.0008	0.0049 ± 0.0009	0.0023 ± 0.0011	0.0414 ± 0.0316	0.0001 ± 0.0000	8.8961 ± 0.2962	84.9889 ± 1.4196	0.0102 ± 0.0066
	MF + Reg	0.0038 ± 0.0004	0.0027 ± 0.0008	0.0041 ± 0.0007	0.0015 ± 0.0009	0.0413 ± 0.0319	0.0001 ± 0.0001	8.9070 ± 0.2978	85.0510 ± 1.4278	0.0090 ± 0.0059
	MFIPS	0.0915 ± 0.0009	0.0691 ± 0.0010	0.0965 ± 0.0009	0.0046 ± 0.0001	0.0769 ± 0.0060	0.3445 ± 0.0062	1.1858 ± 0.2630	87.5519 ± 1.3441	0.0385 ± 0.0003
	MFIPS-AT	0.4332 ± 0.0032	0.3838 ± 0.0032	0.7429 ± 0.0044	0.4045 ± 0.0055	1.0680 ± 0.1220	0.2236 ± 0.0037	10.0000 ± 0.000	93.7016 ± 0.0133	0.0325 ± 0.0005
	EqBal-RS	0.0030 ± 0.0003	0.0031 ± 0.0001	0.0029 ± 0.0002	0.0002 ± 0.0002	0.0466 ± 0.0104	0.0003 ± 0.0000	8.4471 ± 0.2006	79.2019 ± 0.7734	0.0007 ± 0.0000

Table 7.2: Training results for different MF approach on real-world datasets. (Note that PP denotes Popularity Parity.)

Testing Results										
Datasets	**Algorithms**	**Overall MSE(↓)**	**MSE on $\mathcal{I}_{\mathcal{P}}$(↓)**	**MSE on $\mathcal{I}_{\mathcal{NP}}$(↓)**	PP(↓)	**ARP@10 (↓)**	**NDCG@10 (↑)**	**NonPop@10**	**NonPop@100**	**t-test(↓)**
MovieLens	MF	1.6639 ± 0.0004	1.7070 ± 0.0070	1.6514 ± 0.0005	0.0557 ± 0.0106	0.0687 ± 0.0042	0.0043 ± 0.0007	7.6312 ± 1.3770	75.8358 ± 4.3349	-0.0177 ± 0.0043
	MF + Reg	1.6480 ± 0.0039	1.6956 ± 0.0068	1.6340 ± 0.0054	0.0608 ± 0.010	0.0391 ± 0.0109	0.0038 ± 0.0030	5.1611 ± 1.5500	83.2096 ± 0.8734	-0.0206 ± 0.0039
	MFIPS	0.7573 ± 0.0033	0.7195 ± 0.0057	0.7595 ± 0.0031	0.0592 ± 0.0046	0.0423 ± 0.0189	0.0048 ± 0.0438	5.1611 ± 1.5523	40.8416 ± 4.9498	0.0031 ± 0.0003
	MFIPS-AT	0.9816 ± 0.0023	0.9366 ± 0.0030	1.0662 ± 0.0046	0.2595 ± 0.0122	0.0423 ± 0.0187	0.0048 ± 0.0438	9.4032 ± 0.5574	78.5119 ± 6.9949	0.0031 ± 0.0003
	EqBal-RS	0.9354 ± 0.0014	0.9219 ± 0.0030	0.9608 ± 0.0025	0.0388 ± 0.0030	0.0090 ± 0.0021	0.0077 ± 0.0028	8.7370 ± 0.1989	67.5122 ± 0.7098	0.0188 ± 0.0013
Yahoo	MF	3.4419 ± 0.0110	3.8215 ± 0.0150	2.9810 ± 0.0118	0.8405 ± 0.0172	0.9579 ± 0.0117	0.0044 ± 0.0007	2.8753 ± 1.6116	56.1244 ± 3.9817	-0.2400 ± 0.0049
	MF + Reg	3.3505 ± 0.0095	3.6710 ± 0.0128	2.9613± 0.0111	0.7096 ± 0.0151	0.9466 ± 0.0111	0.0047 ± 0.0007	1.6726 ± 0.0810	56.8050 ± 4.0891	-0.2400 ± 0.0049
	MFIPS	1.5086 ± 0.0229	1.6944 ± 0.0432	1.2783 ± 0.0071	1.2383 ± 0.1308	0.2988 ± 0.0106	0.1883 ± 0.0535	1.0693 ± 1.4405	45.3874 ± 10.0159	-0.0109 ± 0.0008
	MFIPS-AT	1.5569 ± 0.0109	1.6712 ± 0.0101	1.4150 ± 0.0145	0.7906 ± 0.0340	0.6081 ± 0.022	0.3105 ± 0.2810	1.2465 ± 1.2201	43.8971 ± 7.1122	-0.0218 ± 0.0010
	EqBal-RS	2.3266 ± 0.0245	2.4808 ± 0.0423	2.1352 ± 0.0101	0.3455 ± 0.0421	0.1951 ± 0.0088	0.0358 ± 0.0014	5.395 ± 0.1212	63.4087 ± 0.6783	-0.1138 ± 0.0080
Amazon	MF	1.5254 ± 0.0152	1.5011 ± 0.0149	1.5320 ± 0.0200	0.0312 ± 0.0270	0.0414 ± 0.0316	0.0001 ± 0.0001	8.8961 ± 0.2962	84.9889 ± 1.4197	0.0087 ± 0.0077
	MF + Reg	1.5465 ± 0.0150	1.5200 ± 0.0136	1.5441 ± 0.0199	0.0330 ± 0.0260	0.0413 ± 0.0319	0.0001 ± 0.0001	8.9070 ± 0.2978	85.0510 ± 1.4279	0.0088 ± 0.0077
	MFIPS	4.2430 ± 0.0147	4.2631 ± 0.0229	4.2384 ± 0.0150	0.2109 ± 0.1743	0.6253 ± 0.0335	0.0556 ± 0.0012	7.4832 ± 0.8909	78.1514 ± 4.5579	0.0426 ± 0.0031
	MFIPS-AT	0.8720 ± 0.0019	0.8192 ± 0.0015	1.2065 ± 0.0067	0.7846 ± 0.0147	0.6253 ± 0.0335	0.0393 ± 0.0032	10.0000 ± 0.000	93.7048 ± 0.0232	0.0403 ± 0.0004
	EqBal-RS	1.2734 ± 0.0108	1.2467 ± 0.0123	1.4418 ± 0.0164	0.1951 ± 0.0207	0.0117 ± 0.0000	0.0002 ± 0.0000	8.4468 ± 0.2000	79.1763 ± 0.7859	0.0598 ± 0.0000

Table 7.3: Testing results for different algorithms on datasets averaged and standard deviation over ten independent runs. (Note that PP denotes Popularity Parity.)

7.5.1 Evaluation of EqBal-RS against Baseline Methods

We begin by comparing EqBal-RS on loss and Popularity Parity. We fix the number of latent factors (k) as 20 across all experiments (consistent with prior literature [371]) and the number of training steps to 100 for MF, MFR, EqBal-RS, and MFIPS. The MFIPS-AT approach requires pre-training steps because of the pseudo-labeling. So, we fine-tune and select to pre-train MFIPS-AT for 250, 650, and 1000 steps for movielens, yahoo, and amazon datasets, respectively. We limit the post-training steps to 50, owning to significant runtime (see runtime analysis). We report the results on the train set in Table 7.2. We can see that improvement in the fairness metric (popularity parity) comes at a slight trade-off in overall mean square error (MSE) loss; thus, exploring a Pareto frontier between fairness and accuracy is an interesting future direction. Overall, we summarize the observations below:

MovieLens

1. The MF and MFR approaches reduce overall loss by emphasizing more on decreasing loss on popular items, resulting in higher Popularity Parity.

2. MFIPS achieves the lowest MSE losses but suffers high variance, resulting in higher Popularity Parity and lower NDCG than MF, and EqBal-RS.

3. MFIPS-AT reduces variance compared to MFIPS but incurs significantly higher losses and performs poorly on fairness metrics such as Popularity Parity, ARP, and has better NDCG than existing debiasing methods such as MFIPS, MFIPS-AT.

4. EqBal-RS achieves the least fairness parity (bias) while having comparable losses with standard MF and MFR approaches. It also performs well on existing fairness metrics, i.e., ARP and NDCG.

Yahoo Music

1. MFIPS results in the lowest mean value for losses but at high variance and increased parity showing the sensitivity of the approach.

2. EqBal-RS achieves losses comparable to MF and MFR, indicating that it is robust to changes in the dataset. Moreover, it performs well on parity metrics, ensuring that losses are not imbalanced over time.

3. MF, MFR, and MFIPS-AT follow a trend similar to movielens.

4. While EqBal-RS and MF exhibit similar performance on Popularity Parity in datasets with few items (yahoo), MF and MFR struggle with parity in datasets that contain a larger number of items.

Amazon GiftCards

1. MF, MFR maintain ARP close to EqBal-RS but achieve poor NDCG scores. This is because the approaches might be recommending an extremely unpopular item.

2. Amazon GiftCard has relatively few items rated by a comparably large number of users; both MFIPS and MFIPS-AT exhibit high losses and parity.

3. EqBal-RS achieves the least loss with significantly lower parity and ARP. Further, it maintains good recommendation quality (NDCG scores) when compared to methods that have low parity, ARP.

Test Results: The results on test data are reported in Table 7.3 and summarized below:

- In the movielens and yahoo datasets, MF, MFR, and MFIPS-AT experience high test losses, while MFIPS achieves the lowest loss values. Although EqBal-RS achieves slightly higher losses than MFIPS, it has the lowest parity, which is desirable since parity can magnify in models over time.

- MF and MFR exhibit comparable loss values than EqBal-RS in the amazon dataset, but both approaches demonstrate high test variance, making EqBal-RS a more efficient option.

7.5.2 Comparison of Non Popular Items in Top-k List

We now conduct an in-depth analysis of the distribution of non-popular items in top-k recommendations. To compare different `SOTA` methods, we report the mean and deviation of the number of non-popular items in the top-10 recommendation list. While the top-10 setting is widely used in the literature [8, 360], we include results for the top-100 recommendation list to facilitate a more comprehensive comparative analysis. The findings are summarized below and reported in Tables 7.2 and 7.3.

Yahoo: As it is the dataset with few non-popular items, it is evident from the results that traditional methods such as MF, MFR, and existing debiasing methods struggle to have an adequate number of non-popular items. On the contrary, our EqBal-RS maintains better representation (around 4 to 6 in top-10) in both train and test results.

MovieLens and Amazon: These datasets enjoy a healthier representation in MF, MFR, and EqBal-RS for top-10 setting. However, in the top-100 setting, it can be observed that these algorithms excessively prioritize non-popular items, potentially introducing a bias towards popular items. On the contrary, EqBal-RS balances both popular and non-popular items count in recommendation lists while maintaining losses and fairness metrics.

We also explore the distribution of non-popular items among users. To this, we plot the sum of items' popularity (rating frequency see Figure 7.2) in each user's top-k recommendation in Figure 7.3 and 7.4 for k value of 10 and 100 respectively. The visualization in top-10 is limited to yahoo and amazon datasets as comparison is easily

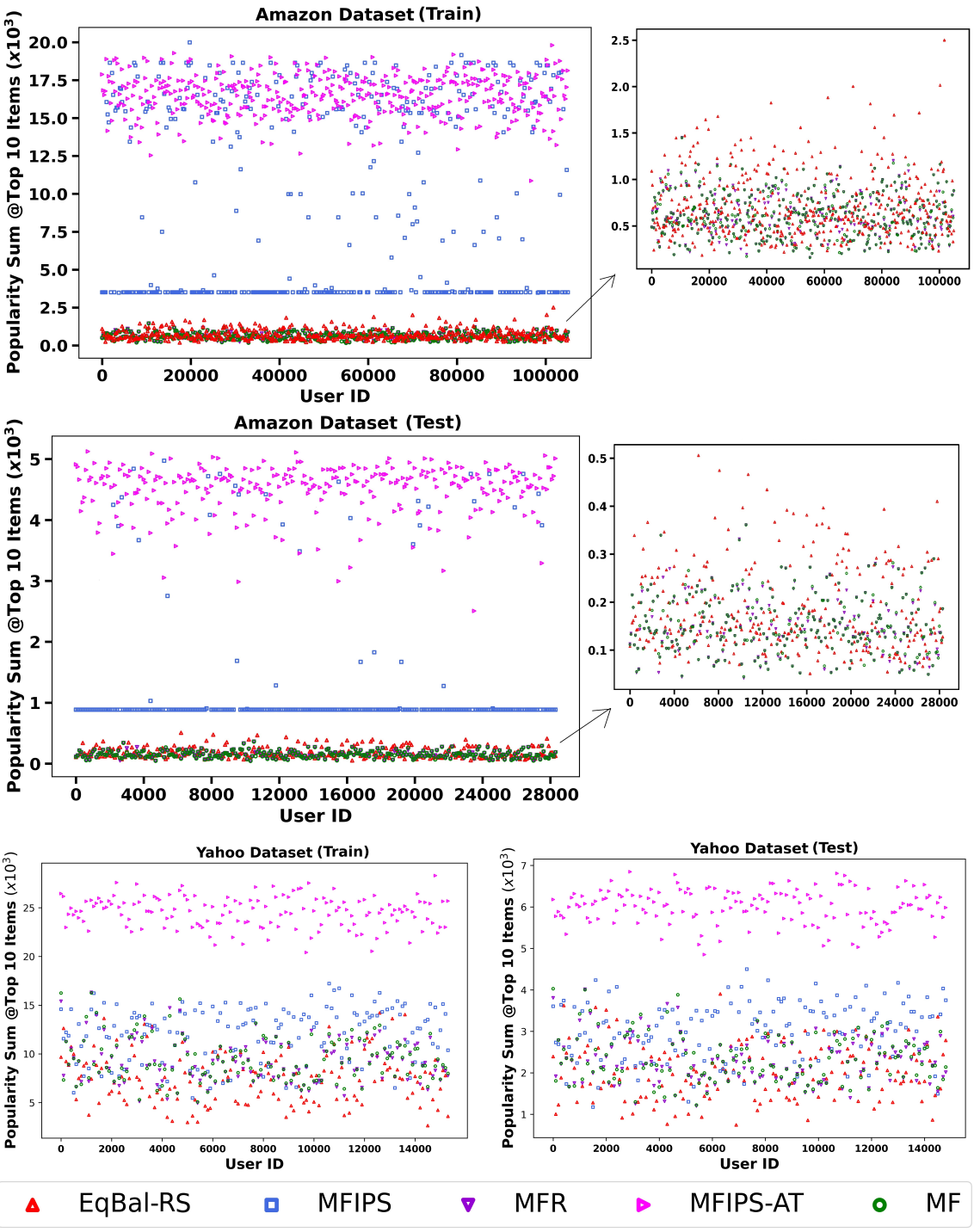


Figure 7.3: Popularity sum of items in top-10 recommendation list across users for different SOTA. (Best viewed in color)

visible considering the total number of ratings, users, and items available. A higher sum indicates that the algorithm primarily recommends popular items, whereas a balanced representation is desirable, so excessively high values are not preferred.

Observations for top-10: MFIPS-AT and MFIPS suffer high popularity sum. In contrast, EqBal-RS maintains a more distributed distribution according to the users' preferences while ensuring a representation of non-popular items. Conversely, MF, MFR

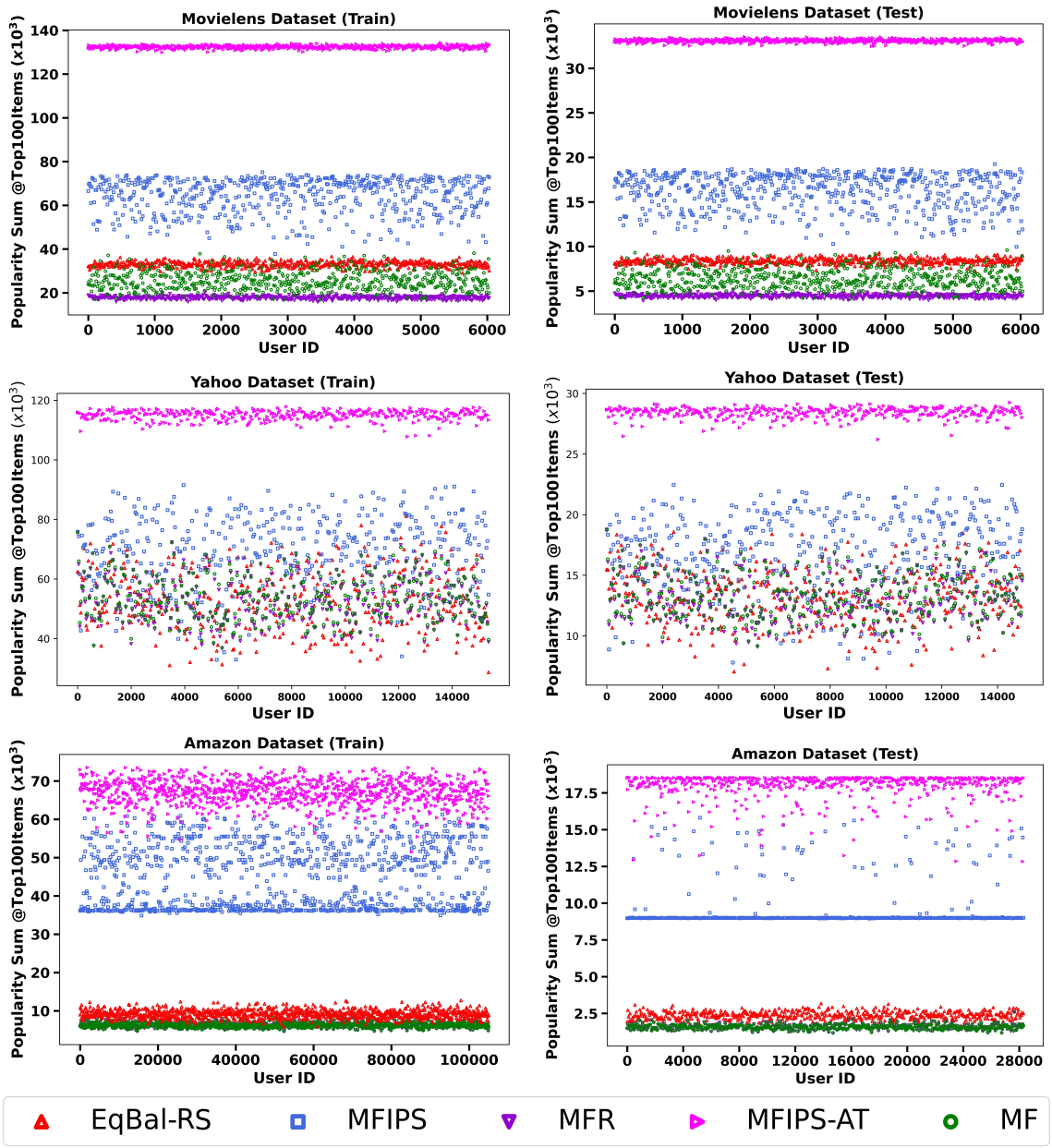


Figure 7.4: Popularity sum of items in top-100 recommendation list across users for different `SOTA`. (Best viewed in color)

is restricted to a small region and has a lower mean value (evident from Table 7.2, 7.3).

Observations for top-100**:** The results are particularly evident in this setting. MFIPS and MFIPS-AT exhibit a high sum, indicating that most of the recommended items come from the popular item segment. MF and MFR, on the other hand, show dataset dependency. In contrast, EqBal-RS maintains a balanced sum value, neither excessively high nor low, regardless of the dataset, thus highlighting the approach's efficacy.

7.5.3 Statistical Significance Testing : t-test

We also compare the difference between the mean error on popular and non-popular items using a t-test that can be computed using Equation 7.11. This analysis helps us understand

the statistical difference between the two groups. A higher t-test value indicates a more significant loss difference between popular and non-popular items. Conversely, smaller values indicate that the losses on both sets of items are more similarly balanced. Our study has large sample sizes, so we have an infinite degree of freedom for the t-test. The train and test set t-test scores are reported in Table 7.2 and 7.3. Evidently, debiasing methods have lower t-test scores than traditional methods such as MF and MFR. Our EqBal-RS has the least t-test on the movielens and amazon datasets, showing the efficacy of our approach compared to `SOTA`.

$$t - test = \frac{L_{\mathcal{NP}} - L_{\mathcal{P}}}{\sqrt{L_{obs}^2 \left(\frac{1}{C_{\mathcal{NP}}} + \frac{1}{C_{\mathcal{P}}}\right)}} \tag{7.11}$$

7.5.4 Runtime Analysis

We report the average runtime of all approaches in Figure 7.5. While MFIPS achieves a lower loss than EqBal-RS, it comes at a considerably high runtime. We further emphasize that MFIPS-AT, even on 50 pre-steps and 50 post-steps, requires double the time than EqBal-RS. Additional pre-training will result in increased overall execution time. Thus, EqBal-RS is a scalable approach for achieving fair and quality recommendations.

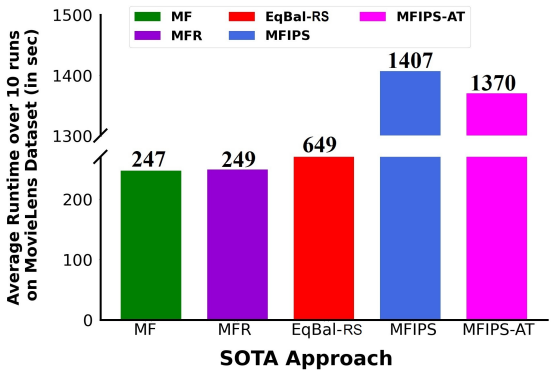


Figure 7.5: Runtime comparison of proposed EqBal-RS against different matrix factorization methods. (Best viewed in color)

7.5.5 Analysis of Training Plots

We now find the answers to the following questions– How well is our model learning? How does the current state of an algorithm change over time? We plot training curves for losses and popularity parity over timesteps to answer these. We train EqBal-RS for ten timesteps, comprising ten epochs of **AdamMF** making of 100 epochs of learning. Similarly, MF, MFR, and MFIPS undergo training for 100 epochs. For comparison with EqBal-RS, we divide these epochs into intervals and plot the results after every ten epochs. Furthermore, as MFIPS-AT requires significant pre-training epochs, we ignore such epochs and plot only the post-training 50 epochs.

The training plots are available in Figure 7.6, with the line depicting the mean value

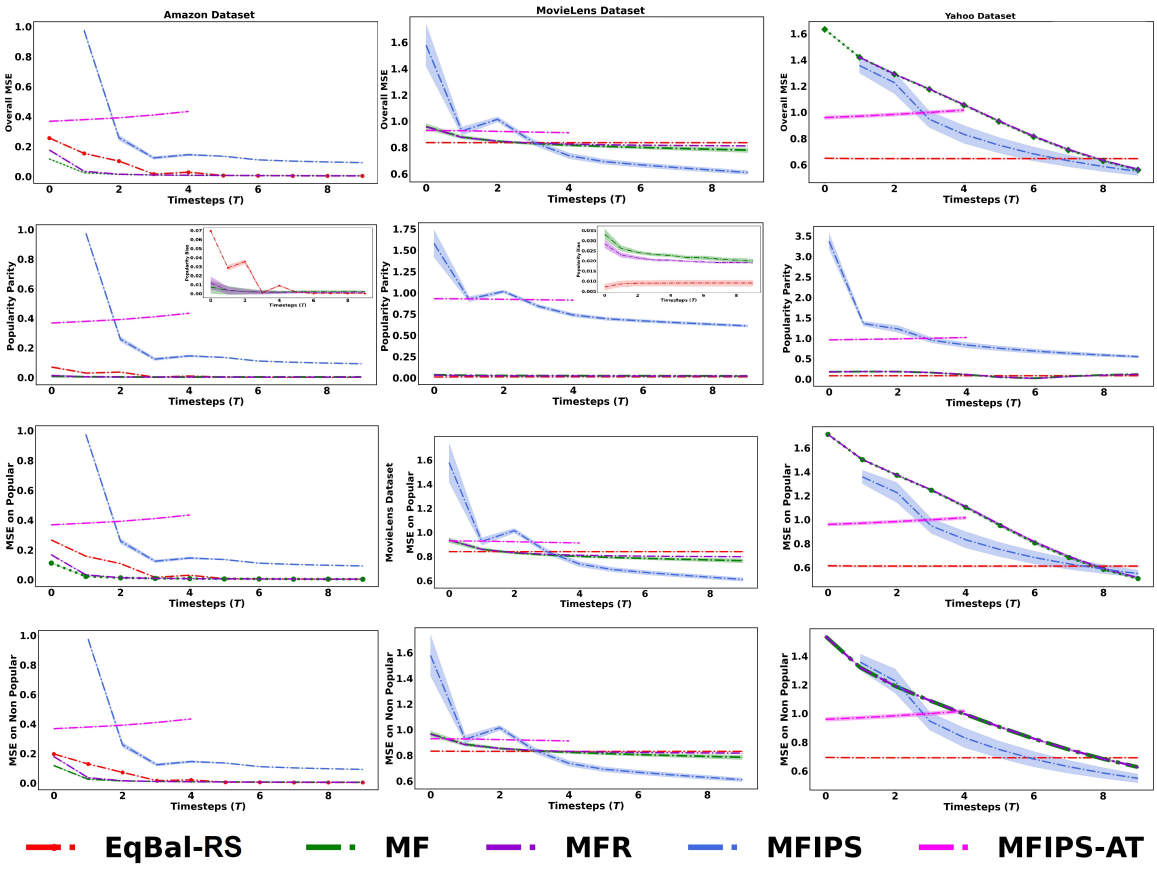


Figure 7.6: Training plots for overall MSE loss (i.e., on $\mathcal{I}$), Popularity Parity, MSE on $\mathcal{I}_{\mathcal{P}}$, MSE on $\mathcal{I}_{\mathcal{NP}}$ on different approaches. (Best viewed in color)

and the shaded region being the standard deviation. To better visualize, we also provide a zoomed view of Popularity Parity. The results for each dataset are summarized below:

–**Amazon**: (1) MFIPS, MFIPS-AT begins with relatively high loss and parity. There is a gradual decrease in values, but it remains much higher than other approaches. (2) EqBal-RS achieves loss quite close to MF, MFR approaches over a few timesteps while maintaining lower parity, showing its efficacy.

–**Movielens**: (1) MFIPS acquires the lowest overall loss but at deteriorated Popularity Parity. (2) EqBal-RS attains the least Popularity Parity while having similar trends and comparable performance on loss values.

–**Yahoo**: (1) All approaches have considerably high loss and parity values over the initial timesteps. (2) EqBal-RS achieves pretty stable and smooth behavior while achieving the best Popularity Parity and comparative loss values.

The plots for MSE on popular and non-popular items follow a similar trend to overall MSE and are illustrated in Figure 7.6.

Ranking based Training Results					
Dataset	**Algorithms**	**NDCG (↑)**	**Dataset**	**Algorithms**	**NDCG (↑)**
MovieLens	MF + FA*IR	0.9661 ± 0.0435	**Yahoo**	MF + FA*IR	1.0000 ± 0.0000
	MF-Reg + FA*IR	0.8011 ± 0.1985		MF-Reg + FA*IR	0.9972 ± 0.0061
	EqBal-RS	0.0204 ± 0.0013		**EqBal-RS**	0.0355 ± 0.0042

Implicit Debiasing Results					
Dataset	**Algorithms**	**Train**		**Test**	
		NDCG (↑)	**ARP (↓)**	**NDCG (↑)**	**ARP (↓)**
MovieLens	CPR	0.0706 ± 0.000	179.8070 ± 0.000	0.0919 ± 0.0000	185.21 ± 0.0000
	EqBal-RS	0.0262 ± 0.0179	0.0185 ± 0.0006	0.0205 ± 0.0014	0.0047 ± 0.0002
Yahoo	CPR	0.0833 ± 0.0000	25.6599 ± 0.0000	0.0869 ± 0.0000	28.325 ± 0.0000
	EqBal-RS	**0.0363 ± 0.0064**	**0.2161 ± 0.0285**	**0.0367 ± 0.0066**	0.0559 ± 0.0111
Amazon	CPR	0.0681 ± 0.0000	17.2408 ± 0.0000	0.0635 ± 0.0000	18.4460 ± 0.0000
	EqBal-RS	**0.000001 ± 0.00**	**0.00001 ± 0.000**	**0.00017 ± 0.000**	0.0007 ± 0.0000

Table 7.4: Results for different algorithms on datasets averaged and standard deviation over ten independent runs.

7.5.6 Comparison with Ranking and Implicit Debiasing Methods

A few recent studies have focused on handling popularity bias in implicit settings. One of the notable current methods is work by Wan et al. [70], which they call CPR. The CPR algorithm enables unbiased recommendations without the need for IPS-based propensity scores. However, it is designed specifically for the implicit and ranking setting. The work sets off the comparison with Borges and Stefanidis [121]. To analyze EqBal-RS in the implicit setting, we convert ratings of 1 to 4 as 0 and explicit ratings of 5 to 1 in line with experimentation in baseline work [70]. The results are reported in Table 7.4.

As CPR is an implicit ranking-based methodology, we limit comparison to well-known ranking metrics, i.e., ARP and NDCG metrics. In EqBal-RS our emphasis is primarily on reducing ARP, which leads to slightly lower performance in NDCG. On the other hand, CPR prioritizes ranking quality (NDCG). Although the performance of EqBal-RS on ARP is reasonably good in the implicit setting, there is still potential for improving ranking quality metrics such as NDCG.

To compare existing post-processing debiasing methods in an implicit setting, we evaluate our algorithm against FA*IR [137] using the NDCG metric. Since FA*IR is a ranking-based methodology, we employ MF and MFR as the underlying models to generate all pairwise user-item ratings (which incurs additional overhead). Our algorithm does not primarily aim at ranking quality in the implicit setting, so observations align with CPR, and we achieve slightly lower efficiency on NDCG. This highlights that EqBal-RS is more suitable for explicit settings. We leave improving performance in the implicit and ranking setting as a potential future direction.

7.5.7 Study on Item Diversity

The percentage of items recommended at least once across all users represents item diversity. Recommending popular items creates a feedback loop, causing diversity to drop over time [370]. EqBal-RS attains a mean value of 0.00012, 4.99×10^{-5}, and 8.49×10^{-6} for yahoo, movielens, and amazon respectively. The negligible deviation (near zero) across runs ensures balanced item visibility.

7.6 Conclusion and Future Work

An appropriate representation of non-popular items is essential for business organizations to give proper visibility to new items. Yet, recommendation engines are well-known to be biased toward popular items. To this, we propose a computationally efficient algorithm – EqBal-RS, that uses a novel metric (Popularity Parity) to measure the popularity bias as differences in losses acquired by non-popular and popular items. EqBal-RS can be particularly useful in real-world scenarios where capturing precise user preferences is crucial. For instance, it can be applied to movie recommendations, where explicit feedback is employed to capture preferences such as genre, plot, and more. Our experiments show EqBal-RS outperforms `SOTA` approaches by reducing popularity bias without affecting the system's overall accuracy and diversity of recommendations. It is worth noting that reducing the Popularity Parity ensures satisfactory performance on existing metrics like ARP and NDCG. However, it is important to note that EqBal-RS is currently restricted to the explicit setting. Although its performance on ARP is reasonably good in the implicit setting, there is room for improvement in ranking quality metrics such as NDCG, which remains a potential direction for future work. Other immediate future direction is tackling popularity bias in multistakeholder systems [372, 373] where the model requires recommendations from various providers and older organizations to generate better customer satisfaction. Apart from this, an extension to deep modeling methods by plugging a DeepMF [374, 375, 376, 377, 378, 379, 380, 381], Nearest Neighbors-based MF [382], and Dynamic MF [383] in place of AdamMF are interesting. One could even investigate the possibility of utilizing distributed [384] or federated MF [385, 386, 387, 388, 389] to deal with massive datasets.

Chapter 8

Conclusion

8.1 Discussion and Open Problems

In this doctoral thesis, we provided different fair algorithms for unsupervised learning. Particularly, we focused on clustering and recommender systems. Figure 8.1, 8.2 provides a brief summary of contributions. Chapters 3 to 6 discussed clustering algorithms in different settings: offline, online and federated. We began by proposing a group fair clustering algorithm for offline setup. We proposed an efficient group fair notion that helped in the development of a polynomial time algorithm for fair clustering. Though both group and individual fairness rose independently in literature, theoretical instances exist in which satisfying both levels of fairness is non-trivial and complementary [64, 65]. However, our study observed that it is possible to satisfy both fairness levels to a reasonable extent on real-world datasets. Motivated by this success in offline clustering, we look into the closely related applications of clustering in facility location problems. We presented an algorithm that achieves good approximation on both group and individual fairness. Next, we extended the idea of satisfying group fairness in clustering to an online setup where the algorithm needs to make irrevocable decisions for each incoming data point. Inspired by individual fairness, we next provided a federated data clustering method that is fair to all participating clients. In Chapter 7, we undertook a slightly different path from clustering and looked into another unsupervised learning technique-recommender systems. Specifically, we looked into matrix factorization and presented an algorithm that outperformed state-of-the-art baselines in achieving lower popularity bias while maintaining overall efficiency and diversity. We now present a summary of the contributions of each major chapter and the direction of future work.

8.1.1 Chapter 3: Group Fair Notion and Algorithms in Offline Clustering

The focus of the chapter was primarily to look into the existing group fairness notions and investigate the prior methods in group fair clustering. We observed that the existing group fairness notions were either limited to binary protected group values or required cluster sizes that are unknown apriori. This led to existing algorithms suffering from large computational or memory requirements or hyper-parameter tuning. We proposed a novel fairness notion that captured the fairness requirements from users for a particular group in terms of the total number of data points from that group value (known apriori). We

further showed that the new notion is a stricter variation of the existing group fairness notion and admits an efficient round-robin algorithm. To this, we proposed two algorithms, namely FRAC$_{OE}$ and FRAC. The FRAC$_{OE}$ algorithm underwent theoretical analysis on cost approximation with respect to optimal clustering and convergence guarantees. The experiments showed that both FRAC$_{OE}$ and FRAC outperformed `SOTA` approaches in objective cost and fairness measures. We also experimentally validated the strictness property of the proposed notion on real-world datasets. A few set of interesting directions for future work are as follows:

1. **Extension to multiple protected groups**: Both proposed methods FRAC and FRAC$_{OE}$ consider non-overlapping protected group identities. That is, the problem considers only one protected group, say race, into account. The work by Bera et al. [12] considered overlapping group identities. For example, in the case of overlapping between gender and race, the protected group can take values such as White-males, White-females, Non-White-males, Non-White-females, etc. The direction of handling multiple multi-valued protected groups has been less explored. In this direction, an initial attempt is undertaken by Bera et al. [12] by incorporating lower (MP) and upper-bound (RD) constraints on the number of data points from each overlapping protected group. For instance, authors incorporate fairness constraints for protected group race and gender separately into the linear program solver. Using a small experimental study, the authors show that in most cases, the fairest solution is when two protected (or sensitive) groups are considered together. Furthermore, their results suggest that the clustering objective cost of including multiple protected groups simultaneously is not too expensive (or far) from a solution cost considering only one protected group. This is an intriguing direction as fairness achieved using one protected group might automatically satisfy fairness requirements for other protected groups or even sometimes degrade the fairness metrics on the other protected group [12].

   Since the work by Bera et al. [12] used linear programming, constraints could be easily incorporated separately for each possible protected group. But in our proposed offline algorithms, we perform a fair assignment procedure (i.e., round-robin rounds in FRAC, FRAC$_{OE}$). Therefore, one possible method to extend current variations for handling different protected groups is to consider each combination of overlapped group values (such as White-males, White-females, Non-White-males, Non-White-females etc.) as a single group value and provide fairness requirements for these as input to the methods. It is important to note that unless the number of protected groups is not too large (holds in real-world scenarios), the technique will not be computationally challenging, but one needs to develop a more efficient strategy with theoretical cost approximation for datasets with a large number of protected groups can be a good direction.

2. **Missing information about protected groups**: There have been recent

developments in supervised machine learning when the information about the protected groups is unknown to the algorithm [187]. The algorithm must identify the sensitive (or protected feature) that can lead to biasness. As of now, no work in group fair clustering considers this challenging problem and can be interesting for future research.

3. **Theoretical guarantees for general** k: We provided theoretical results of two approximation cost guarantees for $k = 2, 3$ and, based on our experimental observation, conjecture the result for any finite k. Current proof techniques for $k \leq 3$ needed intricate case-by-case analysis, which becomes intractable for larger k. Devising better proofing strategies can be a possible direction.

4. **Theoretical guarantees for FRAC**: Our present theoretical analysis is restricted to post-processing FRAC_{OE} method. The analysis for FRAC needs to work on handling the dependency between data points assigned in each cycle round-robin allocation. We leave this as an intriguing research trajectory.

8.1.2 Chapter 4: Balancing Fairness and Efficiency via Novel Welfare Perspective

This chapter primarily focused on handling multiple fairness levels in unsupervised learning. We considered the problem of satisfying group and individual fairness in facility location. Most of the existing works in facility location problems focused on handling utilitarian or egalitarian objectives. However, we modelled the problem using Nash social welfare and proposed `FAIRLOC`. The method helped in satisfying group fairness while simultaneously achieving a good approximation of individual fairness. We also provided cost approximation guarantees and validated `FAIRLOC`'s efficacy on the US census dataset with road map distances. We now provide some interesting future directions below:

1. **Rational behaviour of clients**: `FAIRLOC` assumed that the agents would behave rationally and follow the planner's designated assignments. However, in the real world, agents might become strategic and greedily concentrate on reducing their costs. In such scenarios, it becomes imperative to make the method robust to manipulative behaviour. There are works that provide strategyproofness property but are limited to settings when facilities can be opened at the agent's own location [390, 391, 213, 392]. Designing the strategy-proof method that provides agents with sufficient incentives to not divert from rational behavior under the presence of an explicit facility opening location set is interesting.

2. **Extension of theoretical guarantees**: Since our proving methodology followed similar lines as proofs for FRAC_{OE}. Providing proofs for k centers can be a possible direction.

3. **Connection between Nash welfare and individual fairness**: In order to handle the problem of achieving good approximation for group and individual

fairness, we formulated facility location as a Nash social welfare problem. We experimentally showed that the bound on individual fairness approximation is not too high. However, establishing a relationship between Nash formulation and the induced level of individual fairness approximation factor can be promising.

4. **Efficient updation strategy**: In the proposed `FAIRLOC`, we updated the next suitable facility opening location using a brute force approach as derivatives of objective function did not lead to any closed form relation. Though the number of facilities k and possible facility opening locations are finite, making brute force tractable and studying more efficient ideas can be interesting.

8.1.3 Chapter 5: Group Fairness as Capacity Constraints in Online Clustering

In this chapter, we looked into handling an endless stream of data points, i.e., an online environment. The algorithm had to make an irrevocable decision for each incoming data point about whether to assign it to the existing set of already opened centers or open it as a new center. Prior works in this direction handled online clustering of data points but can result in the formation of highly skewed clusters. In order to handle this challenge, we additionally added capacity constraints for each cluster center. We provided an online algorithm to tackle capacity constraints in h-dimensional space for k-means or k-median. We employed the doubling trick to estimate the number of total incoming data points and used the coupon collector problem to better estimate the initial number of data points that need to be opened as centers. We further extended the method to provide separate capacity constraints for each protected group value in every cluster center. The experimental results validated the performance of both proposed algorithms on the number of centers opened and cost approximation factors. We now summarize the directions that can be undertaken as part of future study:

1. **Robustness to noisy data points**: Online clustering has been under investigation in the presence of noisy data containing outliers [39]. Devising robust methods that help prevent the opening of outlier data points as centers under the presence of capacity constraints is a potential future work. Also, investigating the changes in the cost approximation factors and bounds on the number of centers open is interesting.

2. **Extension to other clustering methods**: Extending the ideas of online clustering beyond k-means and k-median is another good direction. Looking into density-based clustering and k-center problems is yet another direction.

3. **Minimizing the misassignments**: Since capacity constraints and group fairness in online streaming can result in different assignments compared to offline counterparts, focusing on minimizing such reassignments is interesting.

4. **Clustering in the presence of buffer**: In many applications, buffer memory can store a few data points. The decision for these data points can be delayed and can be

taken together in batches. Investigating the problem of satisfying anytime Balance guarantees can be a good study.

8.1.4 Chapter 6: Algorithms for Efficient and Fair Federated Data Clustering

This chapter focused on solving the data clustering problem when data points are scattered across different sites (or servers). We proposed a multi-shot approach that computes clustering without depending on the data distribution across clients. Also, we provided bounds on the quality of centers obtained and showed that if a sufficient number of clients are available, then one can limit the amount of information to be shared between clients and servers. Next, we extended the algorithm to propose a personalized federated algorithm that takes a single round of communication, is independent of data distribution and achieves lower mean per-point objective cost across all clients, thus ensuring that no clients suffer poorly on the centers computed. The method allowed fine-tuning of the global centers on the local dataset to provide a personalized experience. We list the possible directions to further improvise the work as follows:

1. **Presence of malicious clients**: Federated clustering involves taking into consideration the local center representations from all clients. So, it demands an urgent need to analyze the performance of existing methods in the presence of malicious clients [339]. Developing a robust mechanism that does not consider corrupted information from clients is a need of an hour. Another important research question is investigating the theoretical bounds on the quality of resulting cluster centers.

2. **Unlearning of client's data**: With the rising concerns about the privacy of data, another line of research has started investigating techniques to unlearn a collection of data points from the machine learning model [341]. Devising a federated clustering strategy that provides assurance to clients about the deletion of their information from clustering needs exploration.

3. **Noisy data**: Clients can contain data points prone to high noise in data capture pipeline [319]. Since federated setup involves not sharing the original data points with the server, testing the use of robust clustering methods at the client level can be an interesting experimental exercise.

4. **Continual learning setup**: The data points that are getting generated at any client might experience a shift in local optimal cluster centers. An intelligent federated clustering system should keep on incrementally acquiring, revising and deleting centers to maintain the best set of current k center [393].

8.1.5 Correlation of Theoretical bounds with the Practical Applications

We provide the following correlation of theoretical bounds with practical application or scalability for each chapter below:

- **(Chapter 3)**: In this chapter, we provide the cost approximation factor for the offline fair clustering algorithm - FRAC$_{OE}$. The proofs hold under any distance metric as long as they obey triangular inequality, symmetry and positiveness (including zero). We do not make any distributional assumptions about the clusters. Therefore, the proofs for FRAC$_{OE}$ hold practically for all small and large datasets. We also validate the theoretical findings experimentally on large datasets such as Diabetes and Census-II, showing that the cost approximation factor of two is well maintained even at large values of k and n. Therefore, we do not see much degradation in objective cost value while having fairness constraints to objective cost without fairness. On the contrary, while our theoretical bounds are loose on k (exponential in nature), we do not observe any significant degradation with varying k values. Therefore, one can think of tightening the upper bound with respect to k. The study also conducts a runtime analysis on varying numbers of clusters (k) and dataset sizes (n).

- **(Chapter 4)**: In this chapter, we provide objective cost guarantees on FAIRLOC where the distance metric satisfies the assumption of triangular inequality. The experimental validation, however, is conducted by considering real roadmap distances between facilities and agent locations (latitude and longitude). Such roadmap distances hold positiveness but may not obey triangular inequality and symmetry (one-way roads). However, our experimental results validate theoretical approximation factors that FAIRLOC is still able to maintain a better (or lower) objective cost than baselines and does not degrade with an increasing number of facilities (or centers) (k).

- **(Chapter 5)**: In this work, we provide bounds on the number of centers opened and objective cost. In particular, we make a common assumption in the online clustering literature that the data points from different clusters and protected groups are arriving in random ordering. This is a standard assumption in most of the online clustering algorithms for practical use since if ordering is adversarial, then Moshkovitz [260] show that even with knowledge of n, one needs to open up $\Omega(n)$ centers to maintain constant cost approximation to optimal clustering. Here, n is the total number of data points in the stream. It should be noted that our theoretical hold even accommodates adversarial ordering. In such cases, the number of initial centers H_k will increase to $\Omega(n)$, given that the probabilities of data ordering are known for coupon collector mapping.

- **(Chapter 6)**: In this work, we particularly assume that the clusters have Gaussian distribution to prove that global centers obtained in a federated setup will not be

far from optimal ones. The Gaussian assumption is a reasonable assumption and is found in many real-world datasets. We also conducted a validation study on synthetic and real-world small and large datasets, showcasing the efficacy of the proposed method on objective cost and fairness metrics.

8.1.6 General Directions for Future Work in Clustering

We summarize the directions for future researchers to investigate as follows:

- **Pareto frontier Analysis** – The Pareto-optimal frontier provides a complete characterization of the trade-off between multiple objectives in an optimization problem. Many current studies in fair clustering consider fairness requirements as hard constraints or provide guarantees based on data-dependent constraints. A study of the Pareto frontier between fairness and clustering objective cost would help theoreticians and practitioners understand situations in which the trade-off is of most practical significance. The extent to which such a characterization is possible and the study of algorithmic frameworks to achieve this trade-off is an interesting open problem.

- **Generalizations to multiple protected groups** – We briefly reviewed generalization of $\boldsymbol{\tau}$-RD and $\boldsymbol{\tau}$-MP fairness notions to multiple overlapping protected groups setting in Section 2.1.1. A similar strategy can be used for extending the online algorithm by providing capacity bounds for each possible group value. Extending other fairness notions to multiple protected groups and understanding the relationship between these notions is an important and practically relevant research direction.

- **Gaming and incentives design** – In this thesis and most of the fair clustering literature, the data generation process is assumed to be noise-free and non-strategic. Seen as a natural extension of *strategic classification* [394] in a clustering framework, *strategic clustering* (See, [395]) has many practical applications. For instance, in a consumer segmentation application where agents have preferences over segments (i.e., clusters) in which they are assigned, may *game* the algorithm by misreporting their data to obtain the desired assignment. This misreporting may result in a significant loss in the objective function, and consequently, the fairness guarantees may fail to hold. Studying incentives to elicit truthful reports and designing robust gaming fair clustering algorithms is an interesting future work.

- **Integrating Interpretability with Fairness**: Developing methods that ensure fair clustering while ensuring that decisions made are more transparent and understandable [396]. That is, designing techniques that provide insights into the influence of fairness in clustering assignments and associated costs. A few preliminary works in interpretable clustering involve using self-interpretable models

such as decision trees [397, 398, 399, 400, 401]. Extending these works to accommodate fairness aspects is interesting future direction.

- **Evolving definition of fairness**: The thesis studies group and individual fairness paradigms with predefined criteria or definitions in the literature. Within group fairness, the thesis proposes a stricter notion and algorithm of group fairness that is primarily motivated by envy-freeness literature in algorithmic game theory and fair division. The developed offline method is inspired by the greedy round-robin procedure in the fair division of indivisible goods among agents with additive valuations [402]. The procedure ensures the allocation returned after round-robin assignments is envy-free up to any good. We use this methodology to distribute data points (goods) among centers (agents) with distance as valuations. Similarly, the online method uses randomized algorithms while ensuring capacity constraints (fairness properties), and the federated clustering uses fine-tuning to develop a more personalized solution that ensures lower cost deviation (individual fairness). Since all areas require substantially new definitions of fairness, we are unsure if the proposed methodology in the thesis can be applied. However, taking ideas from our work and extending them to the above areas will form interesting future work.

- **Unsupervised Feature Selection**: Clustering has been used as one of the potential feature selection techniques in a few past literature [403, 404]. The primary intuition behind such methods is that a good subset of features is highly correlated to class labels (known or unknown) compared to other non-relevant features. These methods usually, instead of clustering data points, involve clustering features into different clusters in terms based on different predefined similarity criteria. Specifically, the criterion helps identify a subset of features that maintain a considerably good level of similarity criteria close to when executed on whole dataset features. An interesting and less explored direction can be finding answers to the following: Can one use fair clustering methods and select a subset of features with balanced similarity criteria for different protected group values? Will such a feature set help achieve fairer solutions than the feature subset selected by unfair clustering? Can one control the amplification of biasness by selecting fair features [403]?

- **Anomaly Detection**: Clustering is also a potential tool in anomaly detection to detect normal examples (or data points) from anomaly examples. Furthermore, in a parallel direction, it is evident from literature in supervised learning that many existing methods mark normal data points from protected/minority groups as anomalies, specifically when the dataset is highly imbalanced in the proportion of majority and minority groups [405, 406]. An interesting direction can be investigating if a similar problem arises when one applies clustering for anomaly detection. To the best of our knowledge, this direction has not been explored in the existing literature, and using fair clustering can help answer many open questions: What is fair? What is risky (anomaly)? How do we handle the trade-off between the

fairness and efficiency of anomaly detectors? Will the movement of some anomalies to a normal cluster or vice versa help?

- **Committee Selection & Job Hiring**: Another potential direction for research can be investigating the application of group fair clustering in committee selection and job hiring. Can one ensure fair representation of committee members or job candidates from different protected group values?

- **Dynamic or Time-varying Data**: In many real-world applications, the underlying data patterns may not be stationary and evolve over time [407, 408]. The thesis primarily focuses on stationary data distribution without concept drifts. However, in many real-world situations, for example, in loan or job hiring over time, it is possible that females might get balanced compared to males, but law amendments might demand strict minimum representation of people from other gender groups. Also, the need can be seen in applications concerning language or disability as sensitive (or protected group) as, over time, it is possible that some new diseases or language immigrants might force the government (or planners) to ensure minimum representation from their protected group type. Therefore, extending ideas from supervised learning [407, 408] to unsupervised clustering (particularly in online and federated where new data keeps on generating) can help prevent models from getting negatively impacted by concept drifts. One will need to look into how to handle obsolete clusters, define possible time windows for merging or deleting old clusters, investigate the need for a dynamic number of target clusters (k), and handle noise (or outliers) by ensuring buffers before updating the original model.

8.1.7 Chapter 7: Mitigating Popularity Bias in Recommender Systems

The chapter focused on another unsupervised learning technique – recommender systems. We primarily focused on the well-known Matrix Factorization (MF) method in explicit feedback settings. Past literature has reported fairness concerns rising in MF methods about favouritism to popular items over non-popular items. To this, we proposed a novel popularity bias metric that measures bias as the difference between losses on non-popular and popular items. We further proposed an efficient algorithm – EqBal-RS that outperforms baseline methods on proposed and existing popularity bias metrics while maintaining the system's overall accuracy. Some good future works include but are not limited to the following:

1. **Extension to multi-stakeholder systems**: Recommender systems usually involve a number of stakeholders, including but not limited to producers and consumers. A study focusing on handling popularity bias and discussing the intricacies of such systems is worth exploring [372, 373].

2. **Deep matrix factorization**: With the success of the performance of deep learning methods in vision and language processing tasks. Recent efforts have been made by

researchers to study deep matrix factorization [374, 375, 376, 377, 378, 379, 380, 381]. Analyzing deep matrix factorization in our proposed algorithm can be promising.

3. **Extension to federated or distributed setup**: In order to handle a large volume of data, distributed or federated learning plays a vital role. Probing distributed MF[384] or federated MF [385, 386, 387, 388, 389] in our methodology can be yet another promising direction.

4. **Pareto Frontier & Bicriteria Approximate Methods**: Recommendation system literature has numerous evident works that study/show this tradeoff and explore the Pareto frontier [409, 410, 411, 412]. Even in our study, improvement in the fairness metric (popularity parity) came at a slight trade-off (increase) in overall mean square error (MSE) loss. Exploring a Pareto frontier between fairness and accuracy is an interesting future direction. Also, recent efforts have been to study approximation factors for the efficiency of binary matrix factorization in past literature ([413] and see references therein). Authors propose $(1 + \epsilon)$ approximation to binary matrix factorization problem where ϵ is the accuracy parameter. However, these works do not deal with fairness aspects and are limited to just approximation studies on accuracy [414]. Another potential direction for investigation is exploring challenges and bounding the bicriteria approximations on accuracy and fairness metrics. This direction has not been yet explored and can be undertaken by researchers.

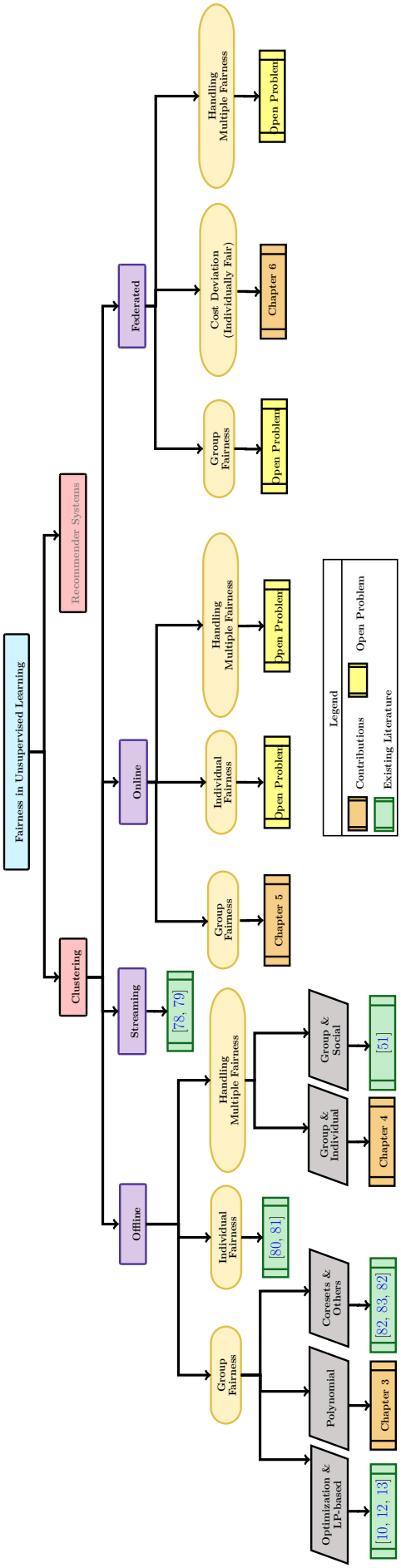


Figure 8.1: A taxonomy of contributions in fair algorithms for clustering.

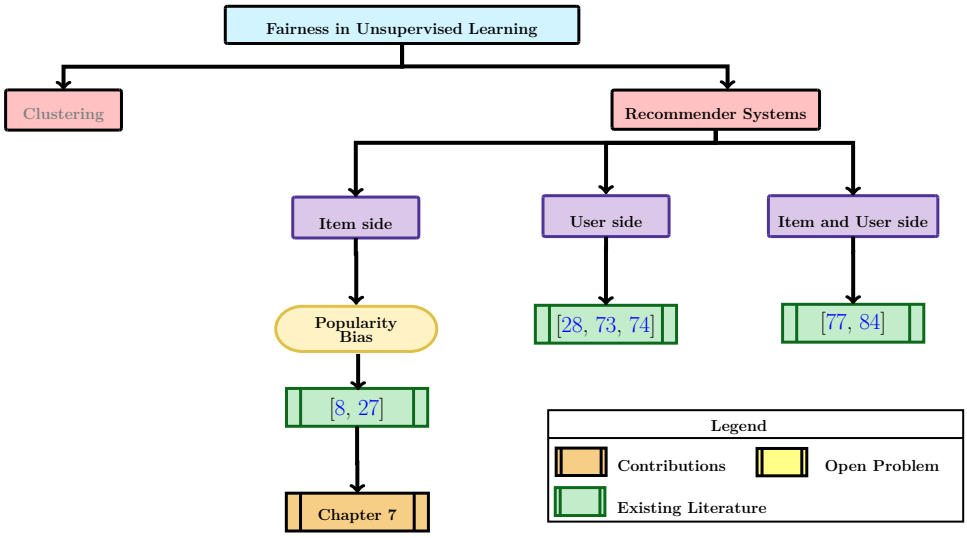


Figure 8.2: A taxonomy of contributions in recommender systems.

References

[1] Herzlich Taylor. Nypost. https://nypost.com/2024/07/29/business/google-search-shows-bias-to-major-brands-pushes-ads-report/, 2024. [Online; accessed 01-November-2024].

[2] Joy Buolamwini and Timnit Gebru. Gender shades: Intersectional accuracy disparities in commercial gender classification. In *Conference on fairness, accountability and transparency*, pages 77–91. PMLR, 2018.

[3] Nedlund Evelina. Cnn business. https://edition.cnn.com/2019/11/12/business/apple-card-gender-bias/index.html, 2019. [Online; accessed 01-November-2024].

[4] Le Chen, Alan Mislove, and Christo Wilson. Peeking beneath the hood of uber. In *Proceedings of the 2015 internet measurement conference*, pages 495–508, 2015.

[5] Lu Donna. Newscientist. https://www.newscientist.com/article/2246202-uber-and-lyft-pricing-algorithms-charge-more-in-non-white-areas/, 2020. [Online; accessed 01-November-2024].

[6] Akshat Pandey and Aylin Caliskan. Disparate impact of artificial intelligence bias in ridehailing economy's price discrimination algorithms. In *Proceedings of the 2021 AAAI/ACM Conference on AI, Ethics, and Society*, pages 822–833, 2021.

[7] A Julia and J Larson. Propublica machine bias. https://www.propublica.org/article/machine-bias-risk-assessments-in-criminal-sentencing, 2016. [Online; accessed 13-August-2021].

[8] Tobias Schnabel, Adith Swaminathan, Ashudeep Singh, Navin Chandak, and Thorsten Joachims. Recommendations as treatments: Debiasing learning and evaluation. In *International Conference on Machine Learning*, pages 1670–1679. PMLR, 2016. doi: 10.48550/arXiv.1602.05352.

[9] Gábor Takács, István Pilászy, Bottyán Németh, and Domonkos Tikk. Matrix factorization and neighbor based algorithms for the netflix prize problem. In *Proceedings of the 2008 ACM Conference on Recommender Systems*, pages 267–274, 2008. doi: 10.1145/1454008.1454049.

[10] Imtiaz Masud Ziko, Jing Yuan, Eric Granger, and Ismail Ben Ayed. Variational fair clustering. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 11202–11209, 2021.

[11] Arturs Backurs, Piotr Indyk, Krzysztof Onak, Baruch Schieber, Ali Vakilian, and Tal Wagner. Scalable fair clustering. In *International Conference on Machine Learning*, pages 405–413. PMLR, 2019.

[12] Suman Bera, Deeparnab Chakrabarty, Nicolas Flores, and Maryam Negahbani. Fair algorithms for clustering. *Advances in Neural Information Processing Systems*, 32, 2019.

[13] Elfarouk Harb and Ho Shan Lam. Kfc: A scalable approximation algorithm for k- center fair clustering. *Advances in Neural Information Processing Systems*, 33: 14509–14519, 2020.

[14] Debajyoti Kar, Sourav Medya, Debmalya Mandal, Arlei Silva, Palash Dey, and Swagato Sanyal. Feature-based individual fairness in k-clustering. *arXiv:2109.04554*, 2021.

[15] Cynthia Dwork, Moritz Hardt, Toniann Pitassi, Omer Reingold, and Richard Zemel. Fairness through awareness. In *Proceedings of the 3rd innovations in theoretical computer science conference*, pages 214–226, 2012.

[16] P Deepak. Whither fair clustering. In *AI for Social Good Workshop. Harvard CRCS*, 2020.

[17] Amir Emad Ghassami, Sajad Khodadadian, and Negar Kiyavash. Fairness in supervised learning: An information theoretic approach. In *IEEE ISIT*, pages 176–180, 2018.

[18] Jose Correa, Andres Cristi, Paul Duetting, and Ashkan Norouzi-Fard. Fairness and bias in online selection. In *International conference on machine learning*, pages 2112–2121. PMLR, 2021.

[19] Bo Li, Lijun Li, Ankang Sun, Chenhao Wang, and Yingfan Wang. Approximate group fairness for clustering. In *International conference on machine learning*, pages 6381–6391. PMLR, 2021.

[20] Joshua K Lee, Yuheng Bu, Deepta Rajan, Prasanna Sattigeri, Rameswar Panda, Subhro Das, and Gregory W Wornell. Fair selective classification via sufficiency. In *ICML*, pages 6076–6086, 2021.

[21] Bram van Berlo, Aaqib Saeed, and Tanir Ozcelebi. Towards federated unsupervised representation learning. In *Proceedings of the third ACM international workshop on edge systems, analytics and networking*, pages 31–36, 2020.

[22] Marcos J. Negreiros, Nelson Maculan, Augusto W.C. Palhano, Albert E.F. Muritiba, and Pablo L.F. Batista. Capacitated clustering models to real-life applications. In Fausto Pedro Garcia Marquez, editor, *Operations Management and Management*

Science, chapter 8. IntechOpen, Rijeka, 2022. doi: 10.5772/intechopen.1000213. URL https://doi.org/10.5772/intechopen.1000213.

[23] Anshika Gupta, Vinay Pant, Sudhanshu Kumar, and Pravesh Kumar Bansal. Bank loan prediction system using machine learning. In *2020 9th International Conference System Modeling and Advancement in Research Trends (SMART)*, pages 423–426. IEEE, 2020.

[24] Mohammad Ahmad Sheikh, Amit Kumar Goel, and Tapas Kumar. An approach for prediction of loan approval using machine learning algorithm. In *2020 international conference on electronics and sustainable communication systems (ICESC)*, pages 490–494. IEEE, 2020.

[25] CK Gomathy, Ms Charulatha, Mr AAkash, and Ms Sowjanya. The loan prediction using machine learning. *International Research Journal of Engineering and Technology (IRJET)*, 8(04), 2021.

[26] Wensi Yang, Yuhang Zhang, Kejiang Ye, Li Li, and Cheng-Zhong Xu. Ffd: A federated learning based method for credit card fraud detection. In *Big Data, Services Conference Federation*. Springer, 2019.

[27] Yuta Saito. Asymmetric tri-training for debiasing missing-not-at-random explicit feedback. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 309–318, 2020. doi: 10.1145/3397271.3401114.

[28] Emre Yalcin and Alper Bilge. Evaluating unfairness of popularity bias in recommender systems: A comprehensive user-centric analysis. *Information Processing & Management*, 59(6):103100, 2022. doi: 10.1016/j.ipm.2022.103100.

[29] Marcy Gordon San Ramon. Ten states sue google for 'anti-competitive' online ad sales. https://brandequity.economictimes.indiatimes.com/news/digital/ten-states-sue-google-for-anti-competitive-online-ad-sales/79771479, 2020.

[30] Amit Saxena, Mukesh Prasad, Akshansh Gupta, Neha Bharill, Om Prakash Patel, Aruna Tiwari, Meng Joo Er, Weiping Ding, and Chin-Teng Lin. A review of clustering techniques and developments. *Neurocomputing*, 267:664–681, 2017.

[31] Anil K Jain, M Narasimha Murty, and Patrick J Flynn. Data clustering: a review. *ACM computing surveys (CSUR)*, 31(3):264–323, 1999.

[32] Mingjun Song and Sanguthevar Rajasekaran. Fast algorithms for constant approximation k-means clustering. *Transactions on Machine Learning and Data Mining*, 3(2):67–79, 2010.

[33] Shi Li and Ola Svensson. Approximating k-median via pseudo-approximation. In *proceedings of the forty-fifth annual ACM symposium on theory of computing*, pages 901–910, 2013.

[34] Dorit S Hochbaum and David B Shmoys. A unified approach to approximation algorithms for bottleneck problems. *Journal of the ACM (JACM)*, 33(3):533–550, 1986.

[35] Teofilo F Gonzalez. Clustering to minimize the maximum intercluster distance. *Theoretical computer science*, 38:293–306, 1985.

[36] Alaettin Zubaroğlu and Volkan Atalay. Data stream clustering: a review. *Artificial Intelligence Review*, 54(2):1201–1236, 2021.

[37] Sudipto Guha, Adam Meyerson, Nina Mishra, Rajeev Motwani, and Liadan O'Callaghan. Clustering data streams: Theory and practice. *IEEE transactions on knowledge and data engineering*, 15(3):515–528, 2003.

[38] Edo Liberty, Ram Sriharsha, and Maxim Sviridenko. An algorithm for online k-means clustering. In *2016 Proceedings of the eighteenth workshop on algorithm engineering and experiments*, pages 81–89. SIAM, 2016.

[39] Aditya Bhaskara and Aravinda Kanchana Ruwanpathirana. Robust algorithms for online k-means clustering. In *Algorithmic Learning Theory*, pages 148–173. PMLR, 2020.

[40] Jie Yan, Jing Liu, Ji Qi, and Zhong-Yuan Zhang. Federated clustering with gan-based data synthesis. *arXiv preprint arXiv:2210.16524*, 2022.

[41] Don Kurian Dennis, Tian Li, and Virginia Smith. Heterogeneity for the win: One-shot federated clustering. In *International Conference on Machine Learning*, pages 2611–2620. PMLR, 2021.

[42] Flavio Chierichetti, Ravi Kumar, Silvio Lattanzi, and Sergei Vassilvitskii. Fair clustering through fairlets. *Advances in neural information processing systems*, 30, 2017.

[43] Shivam Gupta, Shweta Jain, Ganesh Ghalme, Narayanan C Krishnan, and Nandyala Hemachandra. Group and individual fairness in clustering algorithms. In *Ethics in Artificial Intelligence: Bias, Fairness and Beyond*, pages 31–51. Springer, 2023.

[44] Shivam Gupta, Ganesh Ghalme, Narayanan C. Krishnan, and Shweta Jain. Efficient algorithms for fair clustering with a new notion of fairness. *Data Mining and Knowledge Discovery*, pages 1–39, 2023.

[45] Weiting Xu, Jie Hu, Shengdong Du, and Yan Yang. K-means clustering with fairness constraints. In *2021 16th International Conference on Intelligent Systems and Knowledge Engineering (ISKE)*, pages 215–222. IEEE, 2021.

[46] Renbo Pan and Caiming Zhong. Fairness first clustering: A multi-stage approach for mitigating bias. *Electronics*, 12(13):2969, 2023.

[47] Mehrdad Ghadiri, Samira Samadi, and Santosh Vempala. Socially fair k-means clustering. In *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*, pages 438–448, 2021.

[48] Yury Makarychev and Ali Vakilian. Approximation algorithms for socially fair clustering. *arXiv:2103.02512*, 2021.

[49] Matthäus Kleindessner, Pranjal Awasthi, and Jamie Morgenstern. Fair k-center clustering for data summarization. In *International Conference on Machine Learning*, pages 3448–3457. PMLR, 2019.

[50] Haris Angelidakis, Adam Kurpisz, Leon Sering, and Rico Zenklusen. Fair and fast k-center clustering for data summarization. In *International Conference on Machine Learning*, pages 669–702. PMLR, 2022.

[51] John Dickerson, Seyed Esmaeili, Jamie H Morgenstern, and Claire Jie Zhang. Doubly constrained fair clustering. *Advances in Neural Information Processing Systems*, 36, 2024.

[52] Suhas Thejaswi, Ameet Gadekar, Bruno Ordozgoiti, and Michal Osadnik. Clustering with fair-center representation: Parameterized approximation algorithms and heuristics. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 1749–1759, 2022.

[53] Matthew Jones, Huy Nguyen, and Thy Nguyen. Fair k-centers via maximum matching. In *International Conference on Machine Learning*, pages 4940–4949. PMLR, 2020.

[54] Xingyu Chen, Brandon Fain, Liang Lyu, and Kamesh Munagala. Proportionally fair clustering. In *International Conference on Machine Learning*, pages 1032–1041. PMLR, 2019.

[55] Haris Aziz, Barton E Lee, Sean Morota Chu, and Jeremy Vollen. Proportionally representative clustering. *arXiv preprint arXiv:2304.13917*, 2023.

[56] Leon Kellerhals and Jannik Peters. Proportional fairness in clustering: A social choice perspective. *arXiv preprint arXiv:2310.18162*, 2023.

[57] Evi Micha and Nisarg Shah. Proportionally fair clustering revisited. In *47th International Colloquium on Automata, Languages, and Programming (ICALP 2020)*. Schloss Dagstuhl-Leibniz-Zentrum für Informatik, 2020.

[58] Bo Li, Lijun Li, Ankang Sun, Chenhao Wang, and Yingfan Wang. Approximate group fairness for clustering. In *ICML*, pages 6381–6391, 2021.

[59] Stanley Simoes, Muiris MacCarthaigh, et al. Cluster-level group representativity fairness in k-means clustering. *arXiv preprint arXiv:2212.14467*, 2022.

[60] Suhas Thejaswi, Bruno Ordozgoiti, and Aristides Gionis. Diversity-aware k-median: Clustering with fair center representation. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 765–780. Springer, 2021.

[61] Deepak and Savitha Sam Abraham. Representativity fairness in clustering. *12th ACM Conference on Web Science*, Jun 2020. URL http://dx.doi.org/10.1145/3394231.3397910.

[62] Mohsen Abbasi, Aditya Bhaskara, and Suresh Venkatasubramanian. Fair clustering via equitable group representations. In *Proceedings of the 2021 ACM conference on fairness, accountability, and transparency*, pages 504–514, 2021.

[63] Brian Brubach, Darshan Chakrabarti, John Dickerson, Samir Khuller, Aravind Srinivasan, and Leonidas Tsepenekas. A pairwise fair and community-preserving approach to k-center clustering. In *International Conference on Machine Learning*, pages 1178–1189. PMLR, 2020.

[64] Nihesh Anderson, Suman K. Bera, Syamantak Das, and Yang Liu. Distributional individual fairness in clustering. *arXiv:2006.12589*, 2020.

[65] Ian Davidson and SS Ravi. Making existing clusterings fairer: Algorithms, complexity results and insights. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 3733–3740, 2020.

[66] Deepjyoti Roy and Mala Dutta. A systematic review and research perspective on recommender systems. *Journal of Big Data*, 9(1):59, 2022.

[67] Ilham Saifudin and Triyanna Widiyaningtyas. Systematic literature review on recommender system: Approach, problem, evaluation techniques, datasets. *IEEE Access*, 2024.

[68] Xavier Amatriain, Josep M Pujol, and Nuria Oliver. I like it... i like it not: Evaluating user ratings noise in recommender systems. In *User Modeling, Adaptation, and Personalization: 17th International Conference, UMAP 2009, formerly UM and AH, Trento, Italy, June 22-26, 2009. Proceedings 17*, pages 247–258. Springer, 2009. doi: 10.1007/978-3-642-02247-0_24.

[69] Gawesh Jawaheer, Martin Szomszor, and Patty Kostkova. Comparison of implicit and explicit feedback from an online music recommendation service. In *Proceedings of the 1st International Workshop on Information Heterogeneity and Fusion in Recommender Systems*, pages 47–51, 2010. doi: 10.1145/1869446.1869453.

[70] Qi Wan, Xiangnan He, Xiang Wang, Jiancan Wu, Wei Guo, and Ruiming Tang. Cross pairwise ranking for unbiased item recommendation. In *Proceedings of the ACM Web Conference 2022*, pages 2370–2378, 2022. doi: 10.1145/3485447.3512010.

[71] Priyanka Gupta, Ankit Sharma, Pankaj Malhotra, Lovekesh Vig, and Gautam Shroff. Causer: Causal session-based recommendations for handling popularity bias. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*, pages 3048–3052, 2021. doi: 10.1145/3459637.3482071.

[72] Jiawei Chen, Hande Dong, Xiang Wang, Fuli Feng, Meng Wang, and Xiangnan He. Bias and debias in recommender system: A survey and future directions. *ACM Transactions on Information Systems*, 41(3):1–39, 2023. doi: 10.1145/3564284.

[73] Yunqi Li, Hanxiong Chen, Zuohui Fu, Yingqiang Ge, and Yongfeng Zhang. User-oriented fairness in recommendation. In *Proceedings of the Web Conference 2021*, pages 624–632, 2021. doi: 10.1145/3442381.3449866.

[74] Emre Yalcin and Alper Bilge. Evaluating unfairness of popularity bias in recommender systems: A comprehensive user-centric analysis. *Information Processing & Management*, 59(6):103100, 2022. doi: 10.1016/j.ipm.2022.103100.

[75] Emre Yalcin and Alper Bilge. Popularity bias in personality perspective: An analysis of how personality traits expose individuals to the unfair recommendation. *Concurrency and Computation: Practice and Experience*, page e7647, 2023. doi: 10.1002/cpe.7647.

[76] Zhongzhou Liu, Yuan Fang, and Min Wu. Mitigating popularity bias for users and items with fairness-centric adaptive recommendation. *ACM Transactions on Information Systems*, 41(3):1–27, 2023. doi: doi/10.1145/3564286.

[77] Mehdi Elahi, Danial Khosh Kholgh, Mohammad Sina Kiarostami, Sorush Saghari, Shiva Parsa Rad, and Marko Tkalčič. Investigating the impact of recommender systems on user-based and item-based popularity bias. *Information Processing & Management*, 58(5):102655, 2021. doi: 10.1016/j.ipm.2021.102655.

[78] Melanie Schmidt, Chris Schwiegelshohn, and Christian Sohler. Fair coresets and streaming algorithms for fair k-means. In *International Workshop on Approximation and Online Algorithms*, pages 232–251. Springer, 2019.

[79] Suman K. Bera, Syamantak Das, Sainyam Galhotra, and Sagar Sudhir Kale. Fair k-center clustering in mapreduce and streaming settings. In *Proceedings of the ACM Web Conference 2022*, WWW '22, page 1414–1422, New York, NY, USA, 2022. Association for Computing Machinery. ISBN 9781450390965. doi: 10.1145/3485447.3512188. URL https://doi.org/10.1145/3485447.3512188.

[80] Sepideh Mahabadi and Ali Vakilian. Individual fairness for k-clustering. In *International conference on machine learning*, pages 6586–6596. PMLR, 2020.

[81] Maryam Negahbani and Deeparnab Chakrabarty. Better algorithms for individually fair k-clustering. *Advances in Neural Information Processing Systems*, 34: 13340–13351, 2021.

[82] Sayan Bandyapadhyay, Fedor V Fomin, and Kirill Simonov. On coresets for fair clustering in metric and euclidean spaces and their applications. *arXiv:2007.10137*, 2020.

[83] Lingxiao Huang, Shaofeng Jiang, and Nisheeth Vishnoi. Coresets for clustering with fairness constraints. *NeurIPS*, pages 7589–7600, 2019.

[84] Ludovico Boratto, Gianni Fenu, and Mirko Marras. Connecting user and item perspectives in popularity debiasing for collaborative recommendation. *Information Processing & Management*, 58(1):102387, 2021. doi: 10.1016/j.ipm.2020.102387.

[85] Di Wu, Qilong Feng, and Jianxin Wang. New approximation algorithms for fair k-median problem. *arXiv preprint arXiv:2202.06259*, 2022.

[86] Mingjun Song and Sanguthevar Rajasekaran. Fast algorithms for constant approximation k-means clustering. *Trans. Mach. Learn. Data Min.*, 3(2):67–79, 2010.

[87] Saeed Sharifi-Malvajerdi, Michael Kearns, and Aaron Roth. Average individual fairness: Algorithms, generalization and experiments. *Advances in neural information processing systems*, 32, 2019.

[88] Christopher Jung, Sampath Kannan, and Neil Lutz. Service in your neighborhood: Fairness in center location. *Foundations of Responsible Computing (FORC)*, 2020.

[89] Darshan Chakrabarti, John P Dickerson, Seyed A Esmaeili, Aravind Srinivasan, and Leonidas Tsepenekas. A new notion of individually fair clustering: α-equitable k-center. *arXiv:2106.05423*, 2021.

[90] Maria-Florina Balcan, Avrim Blum, and Santosh Vempala. A discriminative framework for clustering via similarity functions. In *Proceedings of the fortieth annual ACM symposium on Theory of computing*, pages 671–680, 2008.

[91] Matthäus Kleindessner, Pranjal Awasthi, and Jamie Morgenstern. A notion of individual fairness for clustering. *arXiv:2006.04960*, 2020.

[92] Sara Ahmadian, Alessandro Epasto, Ravi Kumar, and Mohammad Mahdian. Clustering without over-representation. *SIGKDD*, page 267–275, 2019. doi: 10.1145/3292500.3330987. URL http://dx.doi.org/10.1145/3292500.3330987.

[93] Savitha Sam Abraham, Deepak Padmanabhan, and Sowmya S Sundaram. Fairness in clustering with multiple sensitive attributes. In *EDBT/ICDT 2020 Joint Conference*, pages 287–298, 2020.

[94] Seyed Esmaeili, Brian Brubach, Aravind Srinivasan, and John Dickerson. Fair clustering under a bounded cost. *Advances in Neural Information Processing Systems*, 34:14345–14357, 2021.

[95] Matteo Böhm, Adriano Fazzone, Stefano Leonardi, and Chris Schwiegelshohn. Fair clustering with multiple colors. *arXiv:2002.07892*, 2020.

[96] Suyun Liu and Luis Nunes Vicente. A stochastic alternating balance k-means algorithm for fair clustering. *arXiv:2105.14172*, 2021.

[97] Ali Vakilian and Mustafa Yalçıner. Improved approximation algorithms for individually fair clustering. *arXiv:2106.14043*, 2021.

[98] Clemens Rösner and Melanie Schmidt. Privacy preserving clustering with constraints. *arXiv:1802.02497*, 2018.

[99] Zvi Galil. Efficient algorithms for finding maximum matching in graphs. *ACM Computing Surveys (CSUR)*, 18(1):23–38, 1986.

[100] Takao Asano and Yasuhito Asano. Recent developments in maximum flow algorithms. *Journal of the Operations Research Society of Japan*, 43(1):2–31, 2000.

[101] Hanan Samet. The quadtree and related hierarchical data structures. *ACM Computing Surveys (CSUR)*, 16(2):187–260, 1984.

[102] Ioana O Bercea, Martin Groß, Samir Khuller, Aounon Kumar, Clemens Rösner, Daniel R Schmidt, and Melanie Schmidt. On the cost of essentially fair clusterings. *arXiv:1811.10319*, 2018.

[103] Jianbo Shi and Jitendra Malik. Normalized cuts and image segmentation. *IEEE Transactions on pattern analysis and machine intelligence*, 22(8):888–905, 2000.

[104] Kenneth Lange and Hua Zhou. A legacy of em algorithms. *International Statistical Review*, 90:S52–S66, 2022.

[105] Moses Charikar, Konstantin Makarychev, and Yury Makarychev. Local global tradeoffs in metric embeddings. *SIAM Journal on Computing*, 39(6):2487–2512, 2010.

[106] T-H Hubert Chan, Michael Dinitz, and Anupam Gupta. Spanners with slack. In *European Symposium on Algorithms*, pages 196–207. Springer, 2006.

[107] Chaitanya Swamy. Improved approximation algorithms for matroid and knapsack median problems and applications. *ACM Trans. Algorithms*, 12(4), aug 2016. ISSN 1549-6325. doi: 10.1145/2963170. URL https://doi.org/10.1145/2963170.

[108] Jaglike Makkar, Bhumika, Shweta Jain, and Shivam Gupta. MFC: A multishot approach to federated data clustering. *European Conference on Artificial Intelligence (ECAI)*, pages 1672 – 1679, 2023.

[109] Pranjal Awasthi and Or Sheffet. Improved spectral-norm bounds for clustering. In *International Workshop on Approximation Algorithms for Combinatorial Optimization*, pages 37–49. Springer, 2012.

[110] Kun Yang, Mohammad Mohammadi Amiri, and Sanjeev R Kulkarni. Greedy centroid initialization for federated k-means. *Knowledge and Information Systems*, pages 1–33, 2024.

[111] Claire Little, Mark Elliot, and Richard Allmendinger. Federated learning for generating synthetic data: a scoping review. *International Journal of Population Data Science*, 8(1), 2023.

[112] Jichan Chung, Kangwook Lee, and Kannan Ramchandran. Federated unsupervised clustering with generative models. In *AAAI 2022 International Workshop on Trustable, Verifiable and Auditable Federated Learning*, volume 4, 2022.

[113] Achintha Wijesinghe, Songyang Zhang, Siyu Qi, and Zhi Ding. Ufed-gan: A secure federated learning framework with constrained computation and unlabeled data. *arXiv preprint arXiv:2308.05870*, 2023.

[114] Jie Yan, Jing Liu, Ji Qi, and Zhong-Yuan Zhang. Privacy-preserving federated deep clustering based on gan. *arXiv preprint arXiv:2211.16965*, 2022.

[115] Songze Li, Sizai Hou, Baturalp Buyukates, and Salman Avestimehr. Secure federated clustering. *arXiv preprint arXiv:2205.15564*, 2022.

[116] WV Leeuw. Bc-fl k-means: A consortium blockchain for federated clustering. *Open Access Theses and Dissertations*, 2022. doi: https://research.tue.nl/nl/studentTheses/0ad42707-ed58-4e59-ae4b-7f2251f8ac24.

[117] Nikhil Ketkar and Nikhil Ketkar. Stochastic gradient descent. *Deep learning with Python: A hands-on introduction*, pages 113–132, 2017.

[118] Himan Abdollahpouri, Masoud Mansoury, Robin Burke, and Bamshad Mobasher. The unfairness of popularity bias in recommendation. *arXiv preprint arXiv:1907.13286*, 2019. doi: 10.48550/arXiv.1907.13286.

[119] Hongzhi Yin, Bin Cui, Jing Li, Junjie Yao, and Chen Chen. Challenging the long tail recommendation. *Proceedings of the VLDB Endowment*, 5(9), 2012. doi: 10.14778/2311906.2311916.

[120] Michael A Hitt. The long tail: Why the future of business is selling less of more, 2007.

[121] Rodrigo Borges and Kostas Stefanidis. on mitigating popularity bias in recommendations via variational autoencoders. In *Proceedings of the 36th Annual ACM Symposium on Applied Computing*, pages 1383–1389, 2021. doi: 10.1145/3412841.3442123.

[122] Himan Abdollahpouri, Robin Burke, and Bamshad Mobasher. Managing popularity bias in recommender systems with personalized re-ranking. *arXiv preprint arXiv:1901.07555*, 2019. doi: 10.48550/arXiv.1901.07555.

[123] Tianxin Wei, Fuli Feng, Jiawei Chen, Ziwei Wu, Jinfeng Yi, and Xiangnan He. Model-agnostic counterfactual reasoning for eliminating popularity bias in recommender system. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, pages 1791–1800, 2021. doi: 10.1145/3447548.3467289.

[124] Emre Yalcin. Blockbuster: A new perspective on popularity-bias in recommender systems. In *2021 6th International Conference on Computer Science and Engineering (UBMK)*, pages 107–112. IEEE, 2021. doi: 10.1109/UBMK52708.2021.9558877.

[125] Sami Khenissi and Olfa Nasraoui. Modeling and counteracting exposure bias in recommender systems. *arXiv preprint arXiv:2001.04832*, 2020. doi: 10.48550/arXiv.2001.04832.

[126] Yang Zhang, Fuli Feng, Xiangnan He, Tianxin Wei, Chonggang Song, Guohui Ling, and Yongdong Zhang. Causal intervention for leveraging popularity bias in recommendation. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 11–20, 2021. doi: 10.1145/3404835.3462875.

[127] Yuta Saito, Suguru Yaginuma, Yuta Nishino, Hayato Sakata, and Kazuhide Nakata. Unbiased recommender learning from missing-not-at-random implicit feedback. In *Proceedings of the 13th International Conference on Web Search and Data Mining*, pages 501–509, 2020. doi: 10.1145/3336191.3371783.

[128] Yu Zheng, Chen Gao, Xiang Li, Xiangnan He, Yong Li, and Depeng Jin. Disentangling user interest and conformity for recommendation with causal embedding. In *Proceedings of the Web Conference 2021*, pages 2980–2991, 2021. doi: 10.1145/3442381.3449788.

[129] Zhihong Chen, Jiawei Wu, Chenliang Li, Jingxu Chen, Rong Xiao, and Binqiang Zhao. Co-training disentangled domain adaptation network for leveraging popularity bias in recommenders. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 60–69, 2022. doi: 10.1145/3477495.3531952.

[130] Ming He, Changshu Li, Xinlei Hu, Xin Chen, and Jiwen Wang. Mitigating popularity bias in recommendation via counterfactual inference. In *International Conference on Database Systems for Advanced Applications*, pages 377–388. Springer, 2022. doi: 10.1007/978-3-031-00129-1_32.

[131] Weijieying Ren, Lei Wang, Kunpeng Liu, Ruocheng Guo, Lim Ee Peng, and Yanjie Fu. Mitigating popularity bias in recommendation with unbalanced interactions: A gradient perspective. In *2022 IEEE International Conference on Data Mining (ICDM)*, pages 438–447. IEEE, 2022. doi: https://doi.ieeecomputersociety.org/10.1109/ICDM54844.2022.00054.

[132] Himan Abdollahpouri and Robin Burke. Reducing popularity bias in recommendation over time. *arXiv preprint arXiv:1906.11711*, 2019. doi: 10.48550/arXiv.1906.11711.

[133] Anastasiia Klimashevskaia, Mehdi Elahi, Dietmar Jannach, Christoph Trattner, and Lars Skjærven. Mitigating popularity bias in recommendation: Potential and limits of calibration approaches. In *International Workshop on Algorithmic Bias in Search and Recommendation*, pages 82–90. Springer, 2022. doi: 10.1007/978-3-031-09316-6_8.

[134] Cataldo Musto, Pasquale Lops, Giovanni Semeraro, et al. Fairness and popularity bias in recommender systems: an empirical evaluation. In *CEUR Workshop PROCEEDINGS*, volume 3078, pages 77–91, 2021.

[135] Arda Antikacioglu and R Ravi. Post processing recommender systems for diversity. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 707–716, 2017. doi: 10.1145/3097983.3098173.

[136] Ninareh Mehrabi, Fred Morstatter, Nripsuta Saxena, Kristina Lerman, and Aram Galstyan. A survey on bias and fairness in machine learning. *ACM Comput. Surv.*, 54(6), jul 2021. ISSN 0360-0300. doi: 10.1145/3457607. URL https://doi.org/10.1145/3457607.

[137] Meike Zehlike, Francesco Bonchi, Carlos Castillo, Sara Hajian, Mohamed Megahed, and Ricardo Baeza-Yates. Fa* ir: A fair top-k ranking algorithm. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*, pages 1569–1578, 2017. doi: 10.1145/3132847.3132938.

[138] Emre Yalcin and Alper Bilge. Investigating and counteracting popularity bias in group recommendations. *Information Processing & Management*, 58(5):102608, 2021. doi: 10.1016/j.ipm.2021.102608.

[139] Qidong Liu, Feng Tian, Qinghua Zheng, and Qianying Wang. Disentangling interest and conformity for eliminating popularity bias in session-based recommendation. *Knowledge and Information Systems*, 65(6):2645–2664, 2023. doi: 10.1007/s10115-023-01839-0.

[140] Sriharsha Dara, C Ravindranath Chowdary, and Chintoo Kumar. A survey on group recommender systems. *Journal of Intelligent Information Systems*, 54(2):271–295, 2020. doi: https://doi.org/10.1007/s10844-018-0542-3.

[141] Shivam Gupta, Ganesh Ghalme, Narayanan C Krishnan, and Shweta Jain. Group fair clustering revisited – notions and efficient algorithm. In *International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, pages 2854–2856, 2023.

[142] Shivam Gupta, Ganesh Ghalme, Narayanan C Krishnan, and Shweta Jain. Group fair clustering revisited – notions and efficient algorithm. *Workshop on Games, Agents and Incentives (GAIW), AAMAS*, 2023.

[143] Alycia N Carey and Xintao Wu. The fairness field guide: Perspectives from social and formal sciences. *arXiv:2201.05216*, 2022.

[144] Eirini Ntoutsi, Pavlos Fafalios, Ujwal Gadiraju, Vasileios Iosifidis, Wolfgang Nejdl, Maria-Esther Vidal, Salvatore Ruggieri, Franco Turini, Symeon Papadopoulos, Emmanouil Krasanakis, et al. Bias in data-driven artificial intelligence systems—an introductory survey. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 10(3):e1356, 2020.

[145] Jeffrey Dastin. Amazon scraps secret ai recruiting tool that showed bias against women. https://www.reuters.com/article/us-amazon-com-jobs-automation-insight-idUSKCN1MK08G, 2018. [Online; accessed 15-August-2021].

[146] Cynthia Dwork, Moritz Hardt, Toniann Pitassi, Omer Reingold, and Richard Zemel. Fairness through awareness. In *Proceedings of the 3rd innovations in theoretical computer science Conference*, pages 214–226, 2012. doi: 10.1145/2090236.2090255.

[147] Yoichi Chikahara, Shinsaku Sakaue, Akinori Fujino, and Hisashi Kashima. Learning individually fair classifier with path-specific causal-effect constraint. In *International and Statistics*, pages 145–153, 2021.

[148] Tai Le Quy, Arjun Roy, Vasileios Iosifidis, Wenbin Zhang, and Eirini Ntoutsi. A survey on datasets for fairness-aware machine learning. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, page e1452, 2022.

[149] Deepak, Joemon M. Jose, and Sanil V. Fairness in unsupervised learning. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*, CIKM '20, page 3511–3512, New York, NY, USA, 2020. Association for Computing Machinery. ISBN 9781450368599. doi: 10.1145/3340531.3412175. URL https://doi.org/10.1145/3340531.3412175.

[150] Anshuman Chhabra, Karina Masalkovaitė, and Prasant Mohapatra. An overview of fairness in clustering. *IEEE Access*, 9:130698–130720, 2021. doi: 10.1109/ACCESS.2021.3114099.

[151] Melanie Schmidt and Julian Wargalla. Coresets for constrained k-median and k-means clustering in low dimensional euclidean space. *arXiv:2106.07319*, 2021.

[152] Sixue Gong, Xiaoming Liu, and Anil K Jain. Mitigating face recognition bias via group adaptive classifier. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3414–3424, 2021.

[153] Wenbin Zhang, Albert Bifet, Xiangliang Zhang, Jeremy C Weiss, and Wolfgang Nejdl. Farf: A fair and adaptive random forests classifier. In *Advances in Knowledge Discovery and Data Mining*, pages 245–256. Springer International Publishing, 2021.

[154] Francesco Ranzato, Caterina Urban, and Marco Zanella. Fair training of decision tree classifiers. *arXiv:2101.00909*, 2021.

[155] Michael Lohaus, Michael Perrot, and Ulrike Von Luxburg. Too relaxed to be fair. In *International Conference on Machine Learning*, pages 6360–6369. PMLR, 2020.

[156] Jaewoong Cho, Gyeongjo Hwang, and Changho Suh. A fair classifier using mutual information. In *2020 IEEE International Symposium on Information Theory (ISIT)*, pages 2521–2526, 2020. doi: 10.1109/ISIT44484.2020.9174293.

[157] Elias Baumann and Josef Lorenz Rumberger. State of the art in fair ML: from moral philosophy and legislation to fair classifiers. *CoRR*, abs/1811.09539, 2018. URL http://arxiv.org/abs/1811.09539.

[158] Stanley Simoes, P Deepak, and Muiris MacCarthaigh. Towards fairer centroids in k-means clustering. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 21583–21591, 2024.

[159] Junyoung Byun and Jaewook Lee. Fast and differentially private fair clustering. In *IJCAI*, pages 5915–5923, 2023.

[160] Ian Davidson, Zilong Bai, Cindy Mylinh Tran, and SS Ravi. Making clusterings fairer by post-processing: algorithms, complexity results and experiments. *Data Mining and Knowledge Discovery*, 37(4):1404–1440, 2023.

[161] Ioana O Bercea, Martin Groß, Samir Khuller, Aounon Kumar, Clemens Rösner, Daniel R Schmidt, and Melanie Schmidt. On the cost of essentially fair clusterings. *arXiv:1811.10319*, 2018.

[162] Rashida Hakim, Ana-Andreea Stoica, Christos H. Papadimitriou, and Mihalis Yannakakis. The fairness-quality trade-off in clustering, 2024. URL https://arxiv.org/abs/2408.10002.

[163] Peng Zhou, Rongwen Li, Zhaolong Ling, Liang Du, and Xinwang Liu. Fair clustering ensemble with equal cluster capacity. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2024.

[164] TH Hubert Chan, Arnaud Guerqin, and Mauro Sozio. Fully dynamic k-center clustering. In *Proceedings of the 2018 World Wide Web Conference*, pages 579–587, 2018.

[165] Tai Le Quy, Arjun Roy, Gunnar Friege, and Eirini Ntoutsi. Fair-capacitated clustering. In *EDM*, 2021.

[166] Jarosław Byrka, Thomas Pensyl, Bartosz Rybicki, Aravind Srinivasan, and Khoa Trinh. An improved approximation for k-median, and positive correlation in budgeted optimization. In *Proceedings of the twenty-sixth annual ACM-SIAM symposium on Discrete algorithms*, pages 737–756. SIAM, 2014.

[167] Seyed Esmaeili, Brian Brubach, Leonidas Tsepenekas, and John Dickerson. Probabilistic fair clustering. *Advances in Neural Information Processing Systems*, 33:12743–12755, 2020.

[168] Sara Ahmadian, Alessandro Epasto, Ravi Kumar, and Mohammad Mahdian. Fair correlation clustering. In Silvia Chiappa and Roberto Calandra, editors, *Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics*, volume 108 of *Proceedings of Machine Learning Research*, pages 4195–4205. PMLR, 26–28 Aug 2020. URL https://proceedings.mlr.press/v108/ahmadian20a.html.

[169] Avishek Bose and William Hamilton. Compositional fairness constraints for graph embeddings. In *International Conference on Machine Learning*, pages 715–724. PMLR, 2019.

[170] Matthäus Kleindessner, Samira Samadi, Pranjal Awasthi, and Jamie Morgenstern. Guarantees for spectral clustering with fairness constraints. In *International Conference on Machine Learning*, pages 3458–3467. PMLR, 2019.

[171] Hongjing Zhang and Ian Davidson. Deep fair discriminative clustering. *arXiv preprint arXiv:2105.14146*, 2021.

[172] Bokun Wang and Ian Davidson. Towards fair deep clustering with multi-state protected variables. *arXiv preprint arXiv:1901.10053*, 2019.

[173] Hanyu Song, Peizhao Li, and Hongfu Liu. Deep clustering based fair outlier detection. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery &; Data Mining*, KDD '21, page 1481–1489, New York, NY, USA, 2021. Association for Computing Machinery. ISBN 9781450383325. doi: 10.1145/3447548.3467225. URL https://doi.org/10.1145/3447548.3467225.

[174] Nihesh Anderson, Suman K Bera, Syamantak Das, and Yang Liu. Distributional individual fairness in clustering. *arXiv:2006.12589*, 2020.

[175] Solon Barocas and Andrew D Selbst. Big data's disparate impact. *CALIFORNIA LAW REVIEW*, pages 671–732, 2016.

[176] Anshuman Chhabra, Adish Singla, and Prasant Mohapatra. Fair clustering using antidote data. *arXiv:2106.00600*, 2021.

[177] Zhili Feng, Praneeth Kacham, and David Woodruff. Dimensionality reduction for the sum-of-distances metric. In *International conference on machine learning*, pages 3220–3229. PMLR, 2021.

[178] Sayan Bandyapadhyay, Tanmay Inamdar, Shreyas Pai, and Kasturi Varadarajan. A constant approximation for colorful k-center. *arXiv:1907.08906*, 2019.

[179] Xinrui Jia, Kshiteej Sheth, and Ola Svensson. Fair colorful k-center clustering. In *International Conference on Integer Programming and Combinatorial Optimization*, pages 209–222. Springer, 2020.

[180] Georg Anegg, Haris Angelidakis, Adam Kurpisz, and Rico Zenklusen. A technique for obtaining true approximations for k-center with covering constraints. In *International Conference on Integer Programming and Combinatorial Optimization*, pages 52–65. Springer, 2020.

[181] Arindam Banerjee and Joydeep Ghosh. Scalable clustering algorithms with balancing constraints. *Data Mining and Knowledge Discovery*, 13(3):365–395, 2006.

[182] Leon Bottou and Yoshua Bengio. Convergence properties of the k-means algorithms. *Advances in neural information processing systems*, 7, 1994.

[183] Shivaram Kalyanakrishnan. k-means clustering. https://www.cse.iitb.ac.in/~shivaram/teaching/old/cs344+386-s2017/resources/classnote-2.pdf, 2016. [Online; accessed 29-May-2022].

[184] Andreas Krause. Clustering and k-means. https://las.inf.ethz.ch/courses/lis-s16/hw/hw4_sol.pdf, 2016. [Online; accessed 29-May-2022].

[185] Hans-Peter Kriegel, Erich Schubert, and Arthur Zimek. The (black) art of runtime evaluation: Are we comparing algorithms or implementations? *Knowl. Inf. Syst.*, 52(2):341–378, aug 2017. ISSN 0219-1377. doi: 10.1007/s10115-016-1004-2. URL https://doi.org/10.1007/s10115-016-1004-2.

[186] Ganesh Ghalme, Vineet Nair, Itay Eilat, Inbal Talgam-Cohen, and Nir Rosenfeld. Strategic classification in the dark. In *International Conference on Machine Learning*, pages 3672–3681. PMLR, 2021.

[187] Sharmila Duppala, Juan Luque, John P Dickerson, and Seyed A Esmaeili. Robust fair clustering with group membership uncertainty sets. *arXiv preprint arXiv:2406.00599*, 2024.

[188] Richard L Church. Alfred weber (1868–1958): The father of industrial location theory and supply-chain design. In *Great Minds in Regional Science, Vol. 2*, pages 89–107. Springer, 2023.

[189] Milda R Saunders, Haena Lee, Chieko Maene, Todd Schuble, and Kathleen A Cagney. Proximity does not equal access: racial disparities in access to high quality dialysis facilities. *Journal of racial and ethnic health disparities*, 1:291–299, 2014.

[190] Heather E Campbell, Laura R Peck, and Michael K Tschudi. Justice for all? a cross-time analysis of toxics release inventory facility location. *Review of Policy Research*, 27(1):1–25, 2010.

[191] Roberto M Fernandez. Race, spatial mismatch, and job accessibility: Evidence from a plant relocation. *Social science research*, 37(3):953–975, 2008.

[192] Michael DiNardi, William L Swann, and Serena Y Kim. Racial/ethnic residential segregation and the availability of opioid and substance use treatment facilities in us counties, 2009–2019. *SSM-population health*, 20:101289, 2022.

[193] Dimitrios Efthymiou. Between meeting quotas and following the duty-bound heart: navigating the formidable dilemma of refugee protection in the eu. *Comparative Migration Studies*, 12(1):26, 2024.

[194] Sara John, Megan R Winkler, Ravneet Kaur, Julia DeAngelo, Alex B Hill, Samantha M Sundermeir, Uriyoan Colon-Ramos, Lucia A Leone, Rachael D Dombrowski, Emma C Lewis, et al. Balancing mission and margins: what makes healthy community food stores successful. *International journal of environmental research and public health*, 19(14):8470, 2022.

[195] Swati Gupta, Jai Moondra, and Mohit Singh. Which lp norm is the fairest? approximations for fair facility location across all" p". In *Proceedings of the 24th ACM Conference on Economics and Computation*, pages 817–817, 2023.

[196] Jim Aisner. https://hbswk.hbs.edu/item/what-went-wrong-at-j-c-penney, 2013. [Online; accessed 10-September-2024].

[197] Katie Le. https://envzone.com/from-legacy-u-s-brand-to-not-so-surprising-case-of-di
-searss-twilight-debunked/, 2023. [Online; accessed 10-September-2024].

[198] Benjamin Rader, Christina M Astley, Kara Sewalk, Paul L Delamater, Kathryn Cordiano, Laura Wronski, Jessica Malaty Rivera, Kai Hallberg, Megan F Pera, Jonathan Cantor, et al. Spatial modeling of vaccine deserts as barriers to controlling sars-cov-2. *Communications Medicine*, 2(1):141, 2022.

[199] Ran Tao, Joni Downs, Theresa M Beckie, Yuzhou Chen, and Warren McNelley. Examining spatial accessibility to covid-19 testing sites in florida. *Annals of GIS*, 26(4):319–327, 2020.

[200] Haena Lee, Julia T Caldwell, Chieko Maene, Kathleen A Cagney, and Milda R Saunders. Racial/ethnic inequities in access to high-quality dialysis treatment in chicago: does neighborhood racial/ethnic composition matter? *Journal of racial and ethnic health disparities*, 7:854–864, 2020.

[201] Siddharth Barman and Mashbat Suzuki. Compatibility of fairness and nash welfare under subadditive valuations. *arXiv preprint arXiv:2407.12461*, 2024.

[202] Alexander Lam, Haris Aziz, and Toby Walsh. Nash welfare and facility location. *arXiv preprint arXiv:2310.04102*, 2023.

[203] Charles S Revelle, Horst A Eiselt, and Mark S Daskin. A bibliography for some fundamental problem categories in discrete location science. *European journal of operational research*, 184(3):817–848, 2008.

[204] G Laport, Stefan Nickel, and Francisco Saldanha da Gama. Location science. cham. *Springer, doi*, 10:978–3, 2015.

[205] Laurence Wolsey, Gérard Cornuéjols, and Georges-L Nemhauser. The uncapacitated facility location problem. 1990.

[206] Fabián A Chudak and David P Williamson. Improved approximation algorithms for capacitated facility location problems. In *International Conference on Integer Programming and Combinatorial Optimization*, pages 99–113. Springer, 1999.

[207] Jozef Kratica, Djordje Dugošija, and Aleksandar Savić. A new mixed integer linear programming model for the multi level uncapacitated facility location problem. *Applied Mathematical Modelling*, 38(7-8):2118–2129, 2014.

[208] Kamal Jain, Mohammad Mahdian, and Amin Saberi. A new greedy approach for facility location problems. In *Proceedings of the thiry-fourth annual ACM symposium on Theory of computing*, pages 731–740, 2002.

[209] Chenhao Wang, Xiaoying Wu, Minming Li, and Hau Chan. Facility's perspective to fair facility location problems. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 5734–5741, 2021.

[210] Chenhao Wang and Mengqi Zhang. Fairness and efficiency in facility location problems with continuous demands. In *Proceedings of the 20th International Conference on Autonomous Agents and MultiAgent Systems*, pages 1371–1379, 2021.

[211] Alexander Lam, Haris Aziz, Bo Li, Fahimeh Ramezani, and Toby Walsh. Proportional fairness in obnoxious facility location. In *Proceedings of the 23rd International Conference on Autonomous Agents and Multiagent Systems*, pages 1075–1083, 2024.

[212] Michael T Marsh and David A Schilling. Equity measurement in facility location analysis: A review and framework. *European journal of operational research*, 74(1): 1–17, 1994.

[213] Jiaqian Li, Minming Li, and Hau Chan. Strategyproof mechanisms for group-fair obnoxious facility location problems. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 9832–9839, 2024.

[214] Houyu Zhou, Miniming Li, and Hau Chan. Strategyproof mechanisms for group-fair facility location problems. In *31st International Joint Conference on Artificial Intelligence and the 25th European Conference on Artificial Intelligence (IJCAI-ECAI 2022)*, pages 613–619. International Joint Conferences on Artificial Intelligence, 2022.

[215] Ron Mosenzon and Ali Vakilian. Scalable algorithms for individual preference stable clustering. In *International Conference on Artificial Intelligence and Statistics*, pages 1108–1116. PMLR, 2024.

[216] Karthik Abinav, Thomas Pensyl, and Bartosz Rybicki. Hardness of facility location problems. 2014.

[217] Shivaram Kalyanakrishnan. K-means clustering. *IIT, Bombay*, 2017.

[218] Opowell. https://math.stackexchange.com/questions/138589/intuition-and-derivation-of-the-geometric-mean, 2015. [Online; accessed 18-Janurary-2025].

[219] Pallavi Jain and Rohit Vaish. Maximizing nash social welfare under two-sided preferences. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 9798–9806, 2024.

[220] Salil Gokhale, Harshul Sagar, Rohit Vaish, and Jatin Yadav. Approximating one-sided and two-sided nash social welfare with capacities. *arXiv preprint arXiv:2411.14007*, 2024.

[221] Alexander Lam. Balancing fairness, efficiency and strategy-proofness in voting and facility location problems. In *Proceedings of the 20th International Conference on Autonomous Agents and MultiAgent Systems*, pages 1818–1819, 2021.

[222] Mohammad-Hossein Bateni, Vincent Cohen-Addad, Alessandro Epasto, and Silvio Lattanzi. A scalable algorithm for individually fair k-means clustering. In *International Conference on Artificial Intelligence and Statistics*, pages 3151–3159. PMLR, 2024.

[223] Shivam Gupta, Shweta Jain, Narayanan C. Krishnan, Ganesh Ghalme, and Nandyala Hemachandra. Online algorithm for clustering with capacity constraints. In *Joint International Conference on Data Science & Management of Data (CODS-COMAD)*, 2024.

[224] Shivam Gupta, Shweta Jain, Narayanan Krishnan, Ganesh Ghalme, and Nandyala Hemachandra. Capacitated online clustering algorithm. *European Conference on Artificial Intelligence*, 2024.

[225] Gbeminiyi John Oyewole and George Alex Thopil. Data clustering: Application and trends. *Artificial Intelligence Review*, 56(7):6439–6475, 2023.

[226] Weibo Lin, Zhu He, and Mingyu Xiao. Balanced clustering: a uniform model and fast algorithm. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence*, pages 2987–2993, 2019.

[227] Junyu Huang, Qilong Feng, Ziyun Huang, Jinhui Xu, and Jianxin Wang. Fls: A new local search algorithm for k-means with smaller search space. In *31st International Joint Conference on Artificial Intelligence, IJCAI 2022*, pages 3092–3098. International Joint Conferences on Artificial Intelligence, 2022.

[228] Absalom E Ezugwu, Abiodun M Ikotun, Olaide O Oyelade, Laith Abualigah, Jeffery O Agushaka, Christopher I Eke, and Andronicus A Akinyelu. A comprehensive survey of clustering algorithms: State-of-the-art machine learning applications, taxonomy, challenges, and future research prospects. *Engineering Applications of Artificial Intelligence*, 110:104743, 2022.

[229] S Geetha, G Poonthalir, and PT Vanathi. Improved k-means algorithm for capacitated clustering problem. *INFOCOMP Journal of Computer Science*, 8(4): 52–59, 2009.

[230] Marcos Negreiros and Augusto Palhano. The capacitated centred clustering problem. *Computers & operations research*, 33(6):1639–1663, 2006.

[231] Derya Dinler and Mustafa Kemal Tural. A survey of constrained clustering. In *Unsupervised learning algorithms*, pages 207–235. Springer, 2016.

[232] Vincent Cohen-Addad and Jason Li. On the fixed-parameter tractability of capacitated clustering. In *46th International Colloquium on Automata, Languages, and Programming (ICALP 2019)*, volume 132, pages 41–1. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, 2019.

[233] Jia Shun Low, Zahra Ghafoori, and Christopher Leckie. Online k-means clustering with lightweight coresets. In *AI 2019: Advances in Artificial Intelligence: 32nd Australasian Joint Conference, Adelaide, SA, Australia, December 2–5, 2019, Proceedings 32*, pages 191–202. Springer, 2019.

[234] Jonathan A Silva, Elaine R Faria, Rodrigo C Barros, Eduardo R Hruschka, André CPLF de Carvalho, and João Gama. Data stream clustering: A survey. *ACM Computing Surveys (CSUR)*, 46(1):1–31, 2013.

[235] Anna Choromanska and Claire Monteleoni. Online clustering with experts. In *Artificial Intelligence and Statistics*, pages 227–235. PMLR, 2012.

[236] Vincent Cohen-Addad, Benjamin Guedj, Varun Kanade, and Guy Rom. Online k-means clustering. In *International Conference on Artificial Intelligence and Statistics*, pages 1126–1134. PMLR, 2021.

[237] Gabriella Divéki and Csanád Imreh. Grid based online algorithms for clustering problems. In *2014 IEEE 15th International Symposium on Computational Intelligence and Informatics (CINTI)*, pages 159–162. IEEE, 2014.

[238] Andrii Dzhoha and Iryna Rozora. Multi-armed bandit problem with online clustering as side information. *Journal of Computational and Applied Mathematics*, 427:115132, 2023.

[239] Gabriella Divéki. Online clustering on the line with square cost variable sized clusters. *Acta Cybernetica*, 21(1):75–88, 2013.

[240] János Csirik, Leah Epstein, Csanád Imreh, and Asaf Levin. Online clustering with variable sized clusters. *Algorithmica*, 65(2):251–274, 2013.

[241] Gabriella Divéki and Csanád Imreh. An online 2-dimensional clustering problem with variable sized clusters. *Optimization and Engineering*, 14(4):575–593, 2013.

[242] John M Mulvey and Michael P Beck. Solving capacitated clustering problems. *European Journal of Operational Research*, 18(3):339–348, 1984.

[243] Feng Mai, Michael J Fry, and Jeffrey W Ohlmann. Model-based capacitated clustering with posterior regularization. *European journal of operational research*, 271(2):594–605, 2018.

[244] Maurizio Boccia, Antonio Sforza, Claudio Sterle, and Igor Vasilyev. A cut and branch approach for the capacitated p-median problem based on fenchel cutting planes. *Journal of mathematical modelling and algorithms*, 7:43–58, 2008.

[245] Mario Gnägi and Philipp Baumann. A matheuristic for large-scale capacitated clustering. *Computers & operations research*, 132:105304, 2021.

[246] Michael K Ng. A note on constrained k-means algorithms. *Pattern Recognition*, 33 (3):515–519, 2000.

[247] Xiaoliang Wu, Qilong Feng, Jinhui Xu, and Jianxin Wang. New algorithms for fair k-center problem with outliers and capacity constraints. *Theoretical Computer Science*, 997:114515, 2024.

[248] Dishant Goyal and Ragesh Jaiswal. Tight fpt approximation for constrained k-center and k-supplier. *Theoretical Computer Science*, 940:190–208, 2023.

[249] Mohammad Hossein Bateni, Aditya Bhaskara, Silvio Lattanzi, and Vahab Mirrokni. Distributed balanced clustering via mapping coresets. *Advances in Neural Information Processing Systems*, 27, 2014.

[250] Hossein Esfandiari, Vahab Mirrokni, and Peilin Zhong. Brief announcement: Streaming balanced clustering. In *Proceedings of the 35th ACM Symposium on Parallelism in Algorithms and Architectures*, pages 311–314, 2023.

[251] Weihua Hu, Takeru Miyato, Seiya Tokui, Eiichi Matsumoto, and Masashi Sugiyama. Learning discrete representations via information maximizing self-augmented training. In *International conference on machine learning*, pages 1558–1567. PMLR, 2017.

[252] Jiabo Huang, Shaogang Gong, and Xiatian Zhu. Deep semantic clustering by partition confidence maximisation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8849–8858, 2020.

[253] Pengxin Zeng, Yunfan Li, Peng Hu, Dezhong Peng, Jiancheng Lv, and Xi Peng. Deep fair clustering via maximizing and minimizing mutual information: Theory, algorithm and metric. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 23986–23995, 2023.

[254] Yunfan Li, Peng Hu, Zitao Liu, Dezhong Peng, Joey Tianyi Zhou, and Xi Peng. Contrastive clustering. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, pages 8547–8555, 2021.

[255] Yunfan Li, Mouxing Yang, Dezhong Peng, Taihao Li, Jiantao Huang, and Xi Peng. Twin contrastive learning for online clustering. *International Journal of Computer Vision*, 130(9):2205–2221, 2022.

[256] Vanessa Tran, Manuel Kammermann, and Philipp Baumann. The mpfcc algorithm: A model-based approach for fair-capacitated clustering. In *2023 IEEE International Conference on Industrial Engineering and Engineering Management (IEEM)*, pages 0677–0681. IEEE, 2023.

[257] Abu Reyan Ahmed, Md Saidur Rahman, and Stephen Kobourov. Online facility assignment. *Theoretical Computer Science*, 806:455–467, 2020.

[258] Matteo Almanza, Flavio Chierichetti, Silvio Lattanzi, Alessandro Panconesi, and Giuseppe Re. Online facility location with multiple advice. *Advances in Neural Information Processing Systems*, 34:4661–4673, 2021.

[259] Aristides V Doumas and Vassilis G Papanicolaou. The coupon collector's problem revisited: asymptotics of the variance. *Advances in Applied Probability*, 44(1): 166–195, 2012.

[260] Michal Moshkovitz. Unexpected effects of online no-substitution k-means clustering. In *Algorithmic Learning Theory*, pages 892–930, 2021.

[261] Moses Charikar, Liadan O'Callaghan, and Rina Panigrahy. Better streaming algorithms for clustering problems. In *Proceedings of the thirty-fifth annual ACM symposium on Theory of computing*, pages 30–39, 2003.

[262] Ke Chen. On coresets for k-median and k-means clustering in metric and euclidean spaces and their applications. *SIAM Journal on Computing*, 39(3):923–947, 2009.

[263] Adam Meyerson. Online facility location. In *Proceedings 42nd IEEE Symposium on Foundations of Computer Science*, pages 426–431. IEEE, 2001.

[264] Charu C Aggarwal, Alexander Hinneburg, and Daniel A Keim. On the surprising behavior of distance metrics in high dimensional space. In *Database theory—ICDT 2001: 8th international conference London, UK, January 4–6, 2001 proceedings 8*, pages 420–434. Springer, 2001.

[265] Alexander Brecko, Erik Kajati, Jiri Koziorek, and Iveta Zolotova. Federated learning for edge computing: A survey. *Applied Sciences*, 12(18):9124, 2022.

[266] James Manyika, Michael Chui, Brad Brown, Jacques Bughin, Richard Dobbs, Charles Roxburgh, and Angela Hung Byers. Big data: The next frontier for innovation, competition, and productivity. 2011.

[267] FY Osisanwo, JET Akinsola, O Awodele, JO Hinmikaiye, O Olakanmi, J Akinjobi, et al. Supervised machine learning algorithms: classification and comparison. *International Journal of Computer Trends and Technology (IJCTT)*, 48(3):128–138, 2017.

[268] Sheng Shen, Tianqing Zhu, Di Wu, Wei Wang, and Wanlei Zhou. From distributed machine learning to federated learning: In the view of data privacy and security. *Concurrency and Computation: Practice and Experience*, 34(16):e6002, 2022.

[269] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. Communication-efficient learning of deep networks from decentralized data. In *Artificial intelligence and statistics*, pages 1273–1282. PMLR, 2017.

[270] Tian Li, Anit Kumar Sahu, Manzil Zaheer, Maziar Sanjabi, Ameet Talwalkar, and Virginia Smith. Federated optimization in heterogeneous networks. *Proceedings of Machine learning and systems*, 2:429–450, 2020.

[271] Tian Li, Shengyuan Hu, Ahmad Beirami, and Virginia Smith. Ditto: Fair and robust federated learning through personalization. In *International conference on machine learning*, pages 6357–6368. PMLR, 2021.

[272] Yuyang Deng, Mohammad Mahdi Kamani, and Mehrdad Mahdavi. Distributionally robust federated averaging. *Advances in neural information processing systems*, 33: 15111–15122, 2020.

[273] Kai Hu, Yaogen Li, Shuai Zhang, Jiasheng Wu, Sheng Gong, Shanshan Jiang, and Liguo Weng. Fedmmd: A federated weighting algorithm considering non-iid and local model deviation. *Expert Systems with Applications*, 237:121463, 2024.

[274] Xing Wu, Jie Pei, Xian-Hua Han, Yen-Wei Chen, Junfeng Yao, Yang Liu, Quan Qian, and Yike Guo. Fedel: Federated ensemble learning for non-iid data. *Expert Systems with Applications*, 237:121390, 2024.

[275] Qiang Yang, Yang Liu, Yong Cheng, Yan Kang, Tianjian Chen, and Han Yu. Federated learning. morgan & claypool publishers, 2019.

[276] Jeffry Wicaksana, Zengqiang Yan, Dong Zhang, Xijie Huang, Huimin Wu, Xin Yang, and Kwang-Ting Cheng. Fedmix: Mixed supervised federated learning for medical image segmentation. *IEEE Transactions on Medical Imaging*, 2022.

[277] Sirui Hu, Ling Feng, Xiaohan Yang, and Yongchao Chen. Fedssc: Shared supervised-contrastive federated learning. *arXiv preprint arXiv:2301.05797*, 2023.

[278] Waqar Muhammad, Maria Mushtaq, Khurum Nazir Junejo, and Muhammad Yaseen Khan. Sentiment analysis of product reviews in the absence of labelled data using supervised learning approaches. *Malaysian Journal of Computer Science*, 33(2): 118–132, 2020.

[279] Lulwah M Al-Harigy, Hana A Al-Nuaim, Naghmeh Moradpoor, and Zhiyuan Tan. Building towards automated cyberbullying detection: A comparative analysis. *Computational Intelligence and Neuroscience*, 2022(1):4794227, 2022.

[280] Rayan Krishnan, Pranav Rajpurkar, and Eric J Topol. Self-supervised learning in medicine and healthcare. *Nature Biomedical Engineering*, 6(12):1346–1352, 2022.

[281] Efthymios Tzinis, Jonah Casebeer, Zhepei Wang, and Paris Smaragdis. Separate but together: Unsupervised federated learning for speech enhancement from non-iid data. In *2021 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, pages 46–50. IEEE, 2021.

[282] Gustavo de Carvalho Bertoli, Lourenço Alves Pereira Junior, Osamu Saotome, and Aldri Luiz dos Santos. Generalizing intrusion detection for heterogeneous networks: A stacked-unsupervised federated learning approach. *Computers & Security*, 127: 103106, 2023.

[283] Dianwen Ng, Xiang Lan, Melissa Min-Szu Yao, Wing P Chan, and Mengling Feng. Federated learning: a collaborative effort to achieve better medical imaging models for individual sites that have small labelled datasets. *Quantitative Imaging in Medicine and Surgery*, 11(2):852, 2021.

[284] Lin Lu, Yao Lin, Yuan Wen, Jinxiong Zhu, and Shengwu Xiong. Federated clustering for recognizing driving styles from private trajectories. *Engineering applications of artificial intelligence*, 118:105714, 2023.

[285] Shivam Gupta, Kirandeep Kaur, and Shweta Jain. Eqbal-RS: Mitigating popularity bias in recommender systems. *Journal of Intelligent Information Systems*, pages 1–26, 2023.

[286] Jayant Vyas, Debasis Das, Santanu Chaudhury, et al. Federated learning based driver recommendation for next generation transportation system. *Expert Systems with Applications*, 225:119951, 2023.

[287] Alexander Viala Bellander and Yazan Ghafir. Towards federated fleet learning leveraging unannotated data. 2023.

[288] Yuwen Zhou, Yuhan Hu, Jing Sun, Rui He, and Wenjie Kang. A semi-federated active learning framework for unlabeled online network data. *Mathematics*, 11(8): 1972, 2023.

[289] Yilun Jin, Yang Liu, Kai Chen, and Qiang Yang. Federated learning without full labels: A survey. *arXiv preprint arXiv:2303.14453*, 2023.

[290] Riccardo Presotto, Gabriele Civitarese, and Claudio Bettini. Federated clustering and semi-supervised learning: a new partnership for personalized human activity recognition. *Pervasive and Mobile Computing*, 88:101726, 2023.

[291] Haowen Lin, Jian Lou, Li Xiong, and Cyrus Shahabi. Semifed: Semi-supervised federated learning with consistency and pseudo-labeling. *arXiv preprint arXiv:2108.09412*, 2021.

[292] Chen Zhang, Zixuan Xie, Bin Yu, Chao Wen, and Yu Xie. Fgss: Federated global self-supervised framework for large-scale unlabeled data. *Applied Soft Computing*, 143:110453, 2023.

[293] Chaoyang He, Zhengyu Yang, Erum Mushtaq, Sunwoo Lee, Mahdi Soltanolkotabi, and Salman Avestimehr. Ssfl: Tackling label deficiency in federated learning via personalized self-supervision. *arXiv preprint arXiv:2110.02470*, 2021.

[294] Vladimír Holỳ, Ondřej Sokol, and Michal Černỳ. Clustering retail products based on customer behaviour. *Applied Soft Computing*, 60:752–762, 2017.

[295] Yue Li, Xiaoquan Chu, Dong Tian, Jianying Feng, and Weisong Mu. Customer segmentation using k-means clustering and the adaptive particle swarm optimization algorithm. *Applied Soft Computing*, 113:107924, 2021.

[296] Zhao-Hui Sun, Tian-Yu Zuo, Di Liang, Xinguo Ming, Zhihua Chen, and Siqi Qiu. Gphc: A heuristic clustering method to customer segmentation. *Applied Soft Computing*, 111:107677, 2021.

[297] Vadisena Venkata Krishna Reddy, Radha Vijaya Kumar Reddy, Masthan Siva Krishna Munaga, Balaji Karnam, Suresh Kumar Maddila, and Chandra Sekhar Kolli. Deep learning-based credit card fraud detection in federated learning. *Expert Systems with Applications*, page 124493, 2024.

[298] Xiaoming Zhang and Lean Yu. Consumer credit risk assessment: A review from the state-of-the-art classification algorithms, data traits, and learning methods. *Expert Systems with Applications*, page 121484, 2023.

[299] Shai Shalev-Shwartz and Shai Ben-David. *Understanding machine learning: From theory to algorithms*. Cambridge university press, 2014.

[300] Yi Wang, Mengshuo Jia, Ning Gao, Leandro Von Krannichfeldt, Mingyang Sun, and Gabriela Hug. Federated clustering for electricity consumption pattern extraction. *IEEE Transactions on Smart Grid*, 13(3):2425–2439, 2022.

[301] Enrico Giampieri. *Unsupervised clustering of MDS data using federated learning.* PhD thesis, Alma Mater Studiorum · University of Bologna, 2022.

[302] Minghao Ye, Junjie Zhang, Zehua Guo, and H Jonathan Chao. Federated traffic engineering with supervised learning in multi-region networks. In *2021 IEEE 29th International Conference on Network Protocols (ICNP)*, pages 1–12. IEEE, 2021.

[303] Sogo Pierre Sanon, Rekha Reddy, Christoph Lipps, and Hans Dieter Schotten. Secure federated learning: An evaluation of homomorphic encrypted network traffic prediction. In *2023 IEEE 20th Consumer Communications & Networking Conference (CCNC)*, pages 1–6. IEEE, 2023.

[304] Hung Du, Srikanth Thudumu, Sankhya Singh, Scott Barnett, Irini Logothetis, Rajesh Vasa, and Kon Mouzakis. Decentralized federated learning strategy with image classification using resnet architecture. In *2023 IEEE 20th Consumer Communications & Networking Conference (CCNC)*, pages 706–707. IEEE, 2023.

[305] Isaac Adjei-Mensah, Xiaoling Zhang, Isaac Osei Agyemang, Sophyani Banaamwini Yussif, Adu Asare Baffour, Bernard Mawuli Cobbinah, Collins Sey, Linda Delali Fiasam, Ijeoma Amuche Chikwendu, and Joseph Roger Arhin. Cov-fed: Federated learning-based framework for covid-19 diagnosis using chest x-ray scans. *Engineering Applications of Artificial Intelligence*, 128:107448, 2024.

[306] Ce Ju, Dashan Gao, Ravikiran Mane, Ben Tan, Yang Liu, and Cuntai Guan. Federated transfer learning for eeg signal classification. In *2020 42nd annual international conference of the IEEE engineering in medicine & biology society (EMBC)*, pages 3040–3045. IEEE, 2020.

[307] Sascha Löbner, Boris Gogov, and Welderufael B Tesfay. Enhancing privacy in federated learning with local differential privacy for email classification. In *International Workshop on Data Privacy Management*, pages 3–18. Springer, 2022.

[308] Sourasekhar Banerjee, Rajiv Misra, Mukesh Prasad, Erik Elmroth, and Monowar H Bhuyan. Multi-diseases classification from chest-x-ray: A federated deep learning approach. In *AI 2020: Advances in Artificial Intelligence: 33rd Australasian Joint Conference, AI 2020, Canberra, ACT, Australia, November 29–30, 2020, Proceedings 33*, pages 3–15. Springer, 2020.

[309] Tzu-Ming Harry Hsu, Hang Qi, and Matthew Brown. Federated visual classification with real-world data distribution. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part X 16*, pages 76–92. Springer, 2020.

[310] Tao Liu, Shouqiang Chen, Meng Wu, and Miao Yu. Federated learning enabled hotel customer classification towards imbalanced data. *Applied Soft Computing*, page 112028, 2024.

[311] Ines Feki, Sourour Ammar, Yousri Kessentini, and Khan Muhammad. Federated learning for covid-19 screening from chest x-ray images. *Applied Soft Computing*, 106:107330, 2021.

[312] Guo-Jun Qi and Jiebo Luo. Small data challenges in big data era: A survey of recent progress on unsupervised and semi-supervised methods. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(4):2168–2187, 2020.

[313] Witold Pedrycz. Federated fcm: Clustering under privacy requirements. *IEEE Transactions on Fuzzy Systems*, 30(8):3384–3388, 2022. doi: 10.1109/TFUZZ.2021.3105193.

[314] Morris Stallmann and Anna Wilbik. Towards federated clustering: A federated fuzzy c-means algorithm (ffcm). *arXiv preprint arXiv:2201.07316*, 2022.

[315] Xingchen Hu, Jindong Qin, Yinghua Shen, Witold Pedrycz, Xinwang Liu, and Jiyuan Liu. An efficient federated multi-view fuzzy c-means clustering method. *IEEE Transactions on Fuzzy Systems*, 2023.

[316] Morris Stallmann and Anna Wilbik. On a framework for federated cluster analysis. *Applied Sciences*, 12(20):10455, 2022.

[317] Yizhang Wang, Wei Pang, Di Wang, and Witold Pedrycz. One-shot federated k-means clustering based on density cores. *Authorea Preprints*, 2023.

[318] Sargam Gupta and Shachi Sharma. Fedclus: Federated clustering from distributed homogeneous data. In *International Conference on Soft Computing and its Engineering Applications*, pages 29–41. Springer, 2022.

[319] Atiq Ur Rehman, Samir Brahim Belhaouari, Tanya Stanko, and Vladimir Gorovoy. Divide to federate clustering concept for unsupervised learning. In *Proceedings of Seventh International Congress on Information and Communication Technology: ICICT 2022, London, Volume 4*, pages 19–29. Springer, 2022.

[320] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[321] Jie Yan, Jing Liu, and Zhong-Yuan Zhang. Ccfc: Bridging federated clustering and contrastive learning. *arXiv preprint arXiv:2401.06634*, 2024.

[322] Jie Yan, Jing Liu, Yi-Zi Ning, and Zhong-Yuan Zhang. Ccfc++: Enhancing federated clustering through feature decorrelation. *arXiv preprint arXiv:2402.12852*, 2024.

[323] Ye Tian, Haolei Weng, and Yang Feng. Unsupervised federated learning: A federated gradient em algorithm for heterogeneous mixture models with robustness against adversarial attacks. *arXiv preprint arXiv:2310.15330*, 2023.

[324] Yue Wu, Shuaicheng Zhang, Wenchao Yu, Yanchi Liu, Quanquan Gu, Dawei Zhou, Haifeng Chen, and Wei Cheng. Personalized federated learning under mixture of distributions. In *International Conference on Machine Learning*, pages 37860–37879. PMLR, 2023.

[325] Liwen Zhang and Zongben Xu. k-pfed: Communication-efficient personalized federated clustering. In *2023 IEEE 3rd International Conference on Electronic Technology, Communication and Information (ICETCI)*, pages 1026–1029. IEEE, 2023.

[326] Md Sirajul Islam, Simin Javaherian, Fei Xu, Xu Yuan, Li Chen, and Nian-Feng Tzeng. Fedclust: Optimizing federated learning on non-iid data through weight-driven client clustering. *arXiv preprint arXiv:2403.04144*, 2024.

[327] Renhao Lu, Weizhe Zhang, Yan Wang, Qiong Li, Xiaoxiong Zhong, Hongwei Yang, and Desheng Wang. Auction-based cluster federated learning in mobile edge computing systems. *IEEE Transactions on Parallel and Distributed Systems*, 34(4): 1145–1158, 2023.

[328] Divyansh Jhunjhunwala, Pranay Sharma, Aushim Nagarkatti, and Gauri Joshi. Fedvarp: Tackling the variance due to partial client participation in federated learning. In *UAI*, pages 906–916, 2022. URL https://proceedings.mlr.press/v180/jhunjhunwala22a.html.

[329] Xiao-Xiang Wei and Hua Huang. Edge devices clustering for federated visual classification: A feature norm based framework. *IEEE Transactions on Image Processing*, 2023.

[330] Younghwan Jeong and Taeyoon Kim. A cluster-driven adaptive training approach for federated learning. *Sensors*, 22(18):7061, 2022.

[331] Ekdeep Singh Lubana, Chi Ian Tang, Fahim Kawsar, Robert P Dick, and Akhil Mathur. Orchestra: Unsupervised federated learning via globally consistent clustering. *arXiv preprint arXiv:2205.11506*, 2022.

[332] Enqi Yu, Zhiwei Ye, Zhiqiang Zhang, Ling Qian, and Meiyi Xie. A federated recommendation algorithm based on user clustering and meta-learning. *Applied Soft Computing*, 158:111483, 2024.

[333] Stuart Lloyd. Least squares quantization in pcm. *IEEE transactions on information theory*, 28(2):129–137, 1982.

[334] Jonas Moritz Kohler and Aurelien Lucchi. Sub-sampled cubic regularization for non-convex optimization. In *ICML*, pages 1895–1904, 2017.

[335] Philippe Flajolet, Daniele Gardy, and Loÿs Thimonier. Birthday paradox, coupon collectors, caching algorithms and self-organizing search. *Discrete Applied Mathematics*, 39(3):207–229, 1992.

[336] Tiandi Ye, Cen Chen, Yinggui Wang, Xiang Li, and Ming Gao. Upfl: Unsupervised personalized federated learning towards new clients. In *Proceedings of the 2024 SIAM International Conference on Data Mining (SDM)*, pages 851–859. SIAM, 2024.

[337] Moses Charikar, Sudipto Guha, Éva Tardos, and David B Shmoys. A constant-factor approximation algorithm for the k-median problem. In *Proceedings of the thirty-first annual ACM symposium on Theory of computing*, pages 1–10, 1999.

[338] Dharmendra S Modha and W Scott Spangler. Feature weighting in k-means clustering. *Machine learning*, 52:217–237, 2003.

[339] Ashish Gupta, George Markowsky, and Sajal K Das. Is performance fairness achievable in presence of attackers under federated learning? In *ECAI 2023*, pages 948–955. IOS Press, 2023.

[340] Ana-Andreea Stoica and Christos Papadimitriou. Strategic clustering, 2018.

[341] Chao Pan, Jin Sima, Saurav Prakash, Vishal Rana, and Olgica Milenkovic. Machine unlearning of federated clusters. *arXiv preprint arXiv:2210.16424*, 2022.

[342] Mohammadmehdi Naghiaei, Hossein A Rahmani, and Mahdi Dehghan. The unfairness of popularity bias in book recommendation. In *Advances in Bias and Fairness in Information Retrieval: Third International Workshop, BIAS 2022, Stavanger, Norway, April 10, 2022, Revised Selected Papers*, pages 69–81. Springer, 2022. doi: 10.1007/978-3-030-45442-5_5.

[343] Dominik Kowald, Markus Schedl, and Elisabeth Lex. The unfairness of popularity bias in music recommendation: A reproducibility study. In *Advances in Information Retrieval: 42nd European Conference on IR Research, ECIR 2020, Lisbon, Portugal, April 14–17, 2020, Proceedings, Part II 42*, pages 35–42. Springer, 2020. doi: 10.1007/978-3-030-45442-5_5.

[344] Bruce Ferwerda, Eveline Ingesson, Michaela Berndl, and Markus Schedl. I don't care how popular you are! investigating popularity bias in music recommendations from a user's perspective. In *Proceedings of the 2023 Conference on Human Information Interaction and Retrieval*, pages 357–361, 2023. doi: 10.1145/3576840.3578287.

[345] Dominik Kowald and Emanuel Lacic. Popularity bias in collaborative filtering-based multimedia recommender systems. In *Advances in Bias and Fairness in Information Retrieval: Third International Workshop, BIAS 2022, Stavanger, Norway, April*

10, 2022, Revised Selected Papers, pages 1–11. Springer, 2022. doi: 10.1007/978-3-031-09316-6_1.

[346] Julián Urbano, Markus Schedl, and Xavier Serra. Evaluation in music information retrieval. *Journal of Intelligent Information Systems*, 41(3):345–369, 2013. doi: https://doi.org/10.1007/s10844-013-0249-4.

[347] Shih-Han Chen, Sok-Ian Sou, and Hsun-Ping Hsieh. Top-n music recommendation framework for precision and novelty under diversity group size and similarity. *Journal of Intelligent Information Systems*, pages 1–26, 2023. doi: https://doi.org/10.1007/s10844-023-00784-2.

[348] Phuong T Nguyen, Riccardo Rubei, Juri Di Rocco, Claudio Di Sipio, Davide Di Ruscio, and Massimiliano Di Penta. Dealing with popularity bias in recommender systems for third-party libraries: How far are we? *arXiv preprint arXiv:2304.10409*, 2023. doi: 10.48550/arXiv.2304.10409.

[349] Li Chen, Marco De Gemmis, Alexander Felfernig, Pasquale Lops, Francesco Ricci, and Giovanni Semeraro. Human decision making and recommender systems. *ACM Transactions on Interactive Intelligent Systems (TIIS)*, 3(3):1–7, 2013. doi: 10.1145/2533670.2533675.

[350] Alexandra Chouldechova and Aaron Roth. A snapshot of the frontiers of fairness in machine learning. *Communications of the ACM*, 63(5):82–89, 2020. doi: 10.1145/3376898.

[351] Catherine Stinson. Algorithms are not neutral: Bias in collaborative filtering. *AI and Ethics*, 2(4):763–770, 2022. doi: 10.1007/s43681-022-00136-w.

[352] Sabrina Karboua, Fouzi Harrag, Farid Meziane, and Amal Boutadjine. Mitigation of popularity bias in recommendation systems. In *Tunisian-Algerian Joint Conference on Applied Computing*, 2022. doi: 10.48550/arXiv.2211.01154.

[353] Dominik Kowald, Gregor Mayr, Markus Schedl, and Elisabeth Lex. *arXiv preprint arXiv:2303.00400*, 2023. doi: 10.1007/978-3-031-37249-0_1.

[354] Diego Carraro and Derek Bridge. A sampling approach to debiasing the offline evaluation of recommender systems. *Journal of Intelligent Information Systems*, pages 1–26, 2022. doi: https://doi.org/10.1007/s10844-021-00651-y.

[355] Anand Konjengbam, Nagendra Kumar, and Manish Singh. Unsupervised tag recommendation for popular and cold products. *Journal of Intelligent Information Systems*, 54:545–566, 2020. doi: https://doi.org/10.1007/s10844-019-00574-9.

[356] Masoud Mansoury, Himan Abdollahpouri, Jessie Smith, Arman Dehpanah, Mykola Pechenizkiy, and Bamshad Mobasher. Investigating potential factors associated with gender discrimination in collaborative recommender systems. In *Proceedings of*

the 33rd International Florida Artificial Intelligence Research Society Conference, FLAIRS 2020, pages 193–196, 2020. doi: 10.48550/arXiv.2002.07786.

[357] Himan Abdollahpouri, Masoud Mansoury, Robin Burke, Bamshad Mobasher, and Edward Malthouse. User-centered evaluation of popularity bias in recommender systems. In *Proceedings of the 29th ACM Conference on User Modeling, Adaptation and Personalization*, pages 119–129, 2021. doi: 10.1145/3450613.3456821.

[358] Jin Huang, Harrie Oosterhuis, and Maarten de Rijke. It is different when items are older: Debiasing recommendations when selection bias and user preferences are dynamic. In *Proceedings of the Fifteenth ACM International Conference on Web Search and Data Mining*, pages 381–389, 2022. doi: 10.1145/3488560.3498375.

[359] Hamidreza Tahmasbi, Mehrdad Jalali, and Hassan Shakeri. TSCMF: Temporal and social collective matrix factorization model for recommender systems. *Journal of Intelligent Information Systems*, 56:169–187, 2021. doi: 10.1007/s10844-020-00613-w.

[360] Kuniaki Saito, Yoshitaka Ushiku, and Tatsuya Harada. Asymmetric tri-training for unsupervised domain adaptation. In *International Conference on Machine Learning*, pages 2988–2997. PMLR, 2017. doi: doi/10.5555/3305890.3305990.

[361] Zohreh Ovaisi, Ragib Ahsan, Yifan Zhang, Kathryn Vasilaky, and Elena Zheleva. Correcting for selection bias in learning-to-rank systems. In *Proceedings of The Web Conference 2020*, pages 1863–1873, 2020. doi: 10.1145/3366423.3380255.

[362] Alejandro Bellogín, Pablo Castells, and Iván Cantador. Statistical biases in information retrieval metrics for recommender systems. *Information Retrieval Journal*, 20:606–634, 2017. doi: 10.1007/s10791-017-9312-z.

[363] Dimitar Nikolov, Mounia Lalmas, Alessandro Flammini, and Filippo Menczer. Quantifying biases in online information exposure. *Journal of the Association for Information Science and Technology*, 70(3):218–229, 2019. doi: 10.1002/asi.24121.

[364] Himan Abdollahpouri, Masoud Mansoury, Robin Burke, and Bamshad Mobasher. The connection between popularity bias, calibration, and fairness in recommendation. In *Proceedings of the 14th ACM Conference on Recommender Systems*, pages 726–731, 2020. doi: 10.1145/3383313.3418487.

[365] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. doi: 10.48550/arXiv.1412.6980.

[366] Himan Abdollahpouri, Masoud Mansoury, Robin Burke, and Bamshad Mobasher. The impact of popularity bias on fairness and calibration in recommendation. *arXiv:1910.05755*, 2019. doi: 10.48550/arXiv.1910.05755.

[367] Yehuda Koren, Robert Bell, and Chris Volinsky. Matrix factorization techniques for recommender systems. *Computer*, 42(8):30–37, 2009. doi: 10.1109/MC.2009.263.

[368] Himan Abdollahpouri, Robin Burke, and Bamshad Mobasher. Controlling popularity bias in learning-to-rank recommendation. In *Proceedings of the Eleventh ACM Conference on Recommender Systems*, pages 42–46, 2017. doi: 10.1145/3109859.3109912.

[369] Kalervo Järvelin and Jaana Kekäläinen. Cumulated gain-based evaluation of ir techniques. *ACM Transactions on Information Systems (TOIS)*, 20(4):422–446, 2002. doi: doi/10.1145/582415.582418.

[370] Ziwei Zhu, Yun He, Xing Zhao, Yin Zhang, Jianling Wang, and James Caverlee. Popularity-opportunity bias in collaborative filtering. In *Proceedings of the 14th ACM International Conference on Web Search and Data Mining*, pages 85–93, 2021. doi: 10.1145/3437963.3441820.

[371] Huy Nguyen and Tien Dinh. A modified regularized non-negative matrix factorization for movielens. In *2012 IEEE RIVF International Conference on Computing & Communication Technologies, Research, Innovation, and Vision for the Future*, pages 1–5. IEEE, 2012. doi: 10.1109/rivf.2012.6169831.

[372] Himan Abdollahpouri, Gediminas Adomavicius, Robin Burke, Ido Guy, Dietmar Jannach, Toshihiro Kamishima, Jan Krasnodebski, and Luiz Pizzato. Beyond personalization: Research directions in multistakeholder recommendation. *arXiv preprint arXiv:1905.01986*, 2019. doi: 10.1007/s11257-019-09256-1.

[373] Hossein A Rahmani, Yashar Deldjoo, Ali Tourani, and Mohammadmehdi Naghiaei. The unfairness of active users and popularity bias in point-of-interest recommendation. In *Advances in Bias and Fairness in Information Retrieval: Third International Workshop, BIAS 2022, Stavanger, Norway, April 10, 2022, Revised Selected Papers*, pages 56–68. Springer, 2022. doi: 10.1007/978-3-031-09316-6_6.

[374] Hong-Jian Xue, Xinyu Dai, Jianbing Zhang, Shujian Huang, and Jiajun Chen. Deep matrix factorization models for recommender systems. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence*, volume 17, pages 3203–3209. Melbourne, Australia, 2017. doi: doi/10.5555/3172077.3172336.

[375] Raúl Lara-Cabrera, Ángel González-Prieto, and Fernando Ortega. Deep matrix factorization approach for collaborative filtering recommender systems. *Applied Sciences*, 10(14):4926, 2020. doi: 10.3390/app10144926.

[376] Mohammed Fadhel Aljunid and Manjaiah Dh. An efficient deep learning approach for collaborative filtering recommender system. *Procedia Computer Science*, 171: 829–836, 2020. doi: 10.1016/j.procs.2020.04.090.

[377] Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua. Neural collaborative filtering. In *Proceedings of the 26th International Conference on World Wide Web*, pages 173–182, 2017. doi: 10.1145/3038912.3052569.

[378] Bam Bahadur Sinha and R Dhanalakshmi. DNN-MF: Deep neural network matrix factorization approach for filtering information in multi-criteria recommender systems. *Neural Computing and Applications*, 34(13):10807–10821, 2022. doi: 10.1007/s00521-022-07012-y.

[379] Gopal Behera and Neeta Nain. DeepNNMF: deep nonlinear non-negative matrix factorization to address sparsity problem of collaborative recommender system. *International Journal of Information Technology*, 14(7):3637–3645, 2022. doi: 10.1007/s41870-022-00982-1.

[380] Huazhen Liu, Wei Wang, Yihan Zhang, Renqian Gu, Yaqi Hao, et al. Neural matrix factorization recommendation for user preference prediction based on explicit and implicit feedback. *Computational Intelligence and Neuroscience*, 2022, 2022. doi: 10.1155/2022/9593957.

[381] Adit Krishnan, Ashish Sharma, Aravind Sankar, and Hari Sundaram. An adversarial approach to improve long-tail performance in neural collaborative filtering. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*, pages 1491–1494, 2018. doi: 10.1145/3269206.3269264.

[382] Edoardo D'Amico, Giovanni Gabbolini, Cesare Bernardis, and Paolo Cremonesi. Analyzing and improving stability of matrix factorization for recommender systems. *Journal of Intelligent Information Systems*, 58(2):255–285, 2022. doi: 10.1007/s10844-021-00686-1.

[383] Zhiyuan Zhang, Yun Liu, Guandong Xu, and Guixun Luo. Recommendation using dmf-based fine tuning method. *Journal of Intelligent Information Systems*, 47:233–246, 2016. doi: 10.1007/s10844-016-0407-6.

[384] Lian Chen, Wangdong Yang, Kenli Li, and Keqin Li. Distributed matrix factorization based on fast optimization for implicit feedback recommendation. *Journal of Intelligent Information Systems*, 56:49–72, 2021. doi: 10.1007/s10844-020-00601-0.

[385] Yong Wang, Mingxing Gao, Xun Ran, Jun Ma, and Leo Yu Zhang. An improved matrix factorization with local differential privacy based on piecewise mechanism for recommendation systems. *Expert Systems with Applications*, 216:119457, 2023. doi: 10.1016/j.eswa.2022.119457.

[386] Vito Walter Anelli, Yashar Deldjoo, Tommaso Di Noia, Antonio Ferrara, and Fedelucio Narducci. User-controlled federated matrix factorization for recommender

systems. *Journal of Intelligent Information Systems*, 58(2):287–309, 2022. doi: 10.1007/s10844-021-00688-z.

[387] Peihua Mai and Yan Pang. Privacy-preserving multi-view matrix factorization for recommender systems. *IEEE Transactions on Artificial Intelligence*, 2023. doi: 10.1109/TAI.2023.3240700.

[388] Maksim E Eren, Luke E Richards, Manish Bhattarai, Roberto Yus, Charles Nicholas, and Boian S Alexandrov. FedSPLIT: one-shot federated recommendation system based on non-negative joint matrix factorization and knowledge distillation. *arXiv preprint arXiv:2205.02359*, 2022. doi: 10.48550/arXiv.2205.02359.

[389] Shuchang Liu, Yingqiang Ge, Shuyuan Xu, Yongfeng Zhang, and Amelie Marian. Fairness-aware federated matrix factorization. In *Proceedings of the 16th ACM Conference on Recommender Systems*, pages 168–178, 2022. doi: 10.1145/3523227.3546771.

[390] Xin Chen, Qizhi Fang, Wenjing Liu, and Yuan Ding. Strategyproof mechanisms for 2-facility location games with minimax envy. In *Algorithmic Aspects in Information and Management: 14th International Conference, AAIM 2020, Jinhua, China, August 10–12, 2020, Proceedings 14*, pages 260–272. Springer, 2020.

[391] Toby Walsh. Strategy proof mechanisms for facility location in euclidean and manhattan space. *arXiv preprint arXiv:2009.07983*, 2020.

[392] Itai Feigenbaum and Jay Sethuraman. Strategyproof mechanisms for one-dimensional hybrid and obnoxious facility location models. In *Workshops at the twenty-ninth AAAI conference on artificial intelligence*, 2015.

[393] Liyuan Wang, Xingxing Zhang, Hang Su, and Jun Zhu. A comprehensive survey of continual learning: theory, method and application. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2024.

[394] Moritz Hardt, Nimrod Megiddo, Christos Papadimitriou, and Mary Wootters. Strategic classification. In *ITCS*, ITCS '16, page 111–122, New York, NY, USA, 2016. Association for Computing Machinery. ISBN 9781450340571.

[395] Ana-Andreea Stoica and Christos Papadimitriou. Strategic clustering. http://www.columbia.edu/~as5001/strategicclustering.pdf, 2018. [Online; accessed 22-January-2022].

[396] Sandhya Saisubramanian, Sainyam Galhotra, and Shlomo Zilberstein. Balancing the tradeoff between clustering value and interpretability. In *Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society*, pages 351–357, 2020.

[397] Nave Frost, Michal Moshkovitz, and Cyrus Rashtchian. Exkmc: Expanding explainable k-means clustering. *arXiv preprint arXiv:2006.02399*, 2020.

[398] Sayan Bandyapadhyay, Fedor V Fomin, Petr A Golovach, William Lochet, Nidhi Purohit, and Kirill Simonov. How to find a good explanation for clustering? *Artificial Intelligence*, 322:103948, 2023.

[399] Konstantin Makarychev and Liren Shan. Explainable k-means: don't be greedy, plant bigger trees! In *Proceedings of the 54th Annual ACM SIGACT Symposium on Theory of Computing*, pages 1629–1642, 2022.

[400] Sanjoy Dasgupta, Nave Frost, Michal Moshkovitz, and Cyrus Rashtchian. Explainable k-means clustering: Theory and practice. In *XXAI Workshop. ICML*, 2020.

[401] Michal Moshkovitz, Sanjoy Dasgupta, Cyrus Rashtchian, and Nave Frost. Explainable k-means and k-medians clustering. In *International conference on machine learning*, pages 7055–7065. PMLR, 2020.

[402] Siddharth Barman, Sanath Kumar Krishnamurthy, and Rohit Vaish. Greedy algorithms for maximizing nash social welfare. In *Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems*, pages 7–13, 2018.

[403] Xiaoying Xing, Hongfu Liu, Chen Chen, and Jundong Li. Fairness-aware unsupervised feature selection. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*, pages 3548–3552, 2021.

[404] Huawen Liu, Xindong Wu, and Shichao Zhang. Feature selection using hierarchical feature clustering. In *Proceedings of the 20th ACM international conference on Information and knowledge management*, pages 979–984, 2011.

[405] Ziwei Wu, Lecheng Zheng, Yuancheng Yu, Ruizhong Qiu, John Birge, and Jingrui He. Fair anomaly detection for imbalanced groups. *arXiv preprint arXiv:2409.10951*, 2024.

[406] Hongjing Zhang and Ian Davidson. Towards fair deep anomaly detection. In *Proceedings of the 2021 ACM conference on fairness, accountability, and transparency*, pages 138–148, 2021.

[407] Wenbin Zhang, Mingli Zhang, Ji Zhang, Zhen Liu, Zhiyuan Chen, Jianwu Wang, Edward Raff, and Enza Messina. Flexible and adaptive fairness-aware learning in non-stationary data streams. In *2020 IEEE 32nd international conference on tools with artificial intelligence (ICTAI)*, pages 399–406. IEEE, 2020.

[408] Diem Pham, Binh Tran, Su Nguyen, Damminda Alahakoon, and Mengjie Zhang. Fairness optimisation with multi-objective swarms for explainable classifiers on data streams. *Complex & Intelligent Systems*, 10(4):4741–4754, 2024.

[409] Kenny Peng, Manish Raghavan, Emma Pierson, Jon Kleinberg, and Nikhil Garg. Reconciling the accuracy-diversity trade-off in recommendations. In *Proceedings of the ACM Web Conference 2024*, pages 1318–1329, 2024.

[410] Lin Xiao, Zhang Min, Zhang Yongfeng, Gu Zhaoquan, Liu Yiqun, and Ma Shaoping. Fairness-aware group recommendation with pareto-efficiency. In *Proceedings of the eleventh ACM conference on recommender systems*, pages 107–115, 2017.

[411] Yuying Zhao, Yu Wang, Yunchao Liu, Xueqi Cheng, Charu C Aggarwal, and Tyler Derr. Fairness and diversity in recommender systems: a survey. *ACM Transactions on Intelligent Systems and Technology*, 16(1):1–28, 2025.

[412] Haolun Wu, Chen Ma, Bhaskar Mitra, Fernando Diaz, and Xue Liu. A multi-objective optimization framework for multi-stakeholder fairness-aware recommendation. *ACM Transactions on Information Systems*, 41(2):1–29, 2022.

[413] Ameya Velingker, Maximilian Vötsch, David Woodruff, and Samson Zhou. Fast $(1+\varepsilon)$-approximation algorithms for binary matrix factorization. In *International Conference on Machine Learning*, pages 34952–34977. PMLR, 2023.

[414] Cho-Jui Hsieh, Nagarajan Natarajan, and Inderjit Dhillon. Pu learning for matrix completion. In *International conference on machine learning*, pages 2445–2453. PMLR, 2015.