

# Empirical Study Of Sampling Heuristics For Fairness In Ranking

by

VINAY MAHARAAJ  
202111024

A Thesis Submitted in Partial Fulfilment of the Requirements for the Degree of

MASTER OF TECHNOLOGY  
in  
INFORMATION AND COMMUNICATION TECHNOLOGY  
to

**DHIRUBHAI AMBANI INSTITUTE OF INFORMATION AND COMMUNICATION TECHNOLOGY**



May, 2023

## Declaration

I hereby declare that

- i) the thesis comprises of my original work towards the degree of Master of Technology in Information and Communication Technology at Dhirubhai Ambani Institute of Information and Communication Technology and has not been submitted elsewhere for a degree,
- ii) due acknowledgment has been made in the text to all the reference material used.

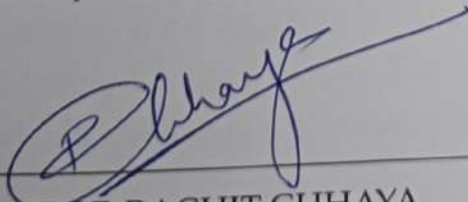
Vinay

---

VINAY MAHARAAJ

## Certificate

This is to certify that the thesis work entitled **Empirical Study Of Sampling Heuristics For Fairness In Ranking** has been carried out by **VINAY MAHARAAJ** for the degree of Master of Technology in Information and Communication Technology at *Dhirubhai Ambani Institute of Information and Communication Technology* under my/our supervision.



---

PROF. RACHIT CHHAYA  
Thesis Supervisor



---

PROF. ARPIT RANA  
Thesis Co-Supervisor

# Acknowledgments

I would like to express my deepest gratitude and appreciation to all those who have supported and guided me throughout the journey of completing this thesis. Their invaluable contributions have played a crucial role in shaping my research and bringing it to fruition.

First and foremost, I am immensely grateful to my thesis guide, **Prof. Rachit Chhaya** and Co-guide **Prof. Arpit Rana**, for their unwavering support, expertise, and guidance. Their insightful suggestions, constructive feedback, and commitment to academic excellence have been instrumental in shaping the direction and quality of this work. I am truly fortunate to have had the opportunity to learn from their vast knowledge and experience.

I would also like to extend my heartfelt thanks to the members of my thesis committee, **Prof. Anuj Tawari** and **Prof. Tathagata Bandhopadhyay**, for their valuable insights, critical feedback, and valuable suggestions that have significantly enhanced the depth and quality of this research. Their expertise and scholarly input have been invaluable in shaping the intellectual rigor of this work.

I am deeply indebted to the faculty members of **DAIICT**, whose teachings and mentorship have enriched my academic journey. Their dedication to fostering a stimulating learning environment and their commitment to excellence have been pivotal in nurturing my intellectual growth and development.

Furthermore, I would like to acknowledge the support received from my mother **Mrs. Babita Devi**, father **Mr. Vijay Maharaj** and my brother **Vineet Maharaj**, their emotional support and sacrifices have been instrumental in helping me overcome challenges and persevere in the face of adversity. I would like to acknowledge the assistance received from my friends **Maulik Sarvaiya** and **Raghav Gorasiya**.

# Contents

<b>Abstract</b>	<b>vi</b>
<b>List of Figures</b>	<b>vii</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Algorithmic Bias . . . . .	5
1.1.1 Presentation Bias . . . . .	5
1.1.2 Evaluation Bias . . . . .	5
1.2 Data Bias . . . . .	5
1.2.1 Statistical Bias . . . . .	5
1.2.2 Pre-existing Bias . . . . .	5
1.2.3 Other Causes . . . . .	6
1.3 Real World Example for Unfairness In Machine Learning . . . . .	6
1.4 Why Fairness is necessary? . . . . .	7
<b>2 Other Related Work</b>	<b>8</b>
2.1 Unfairness in Ranking: An Example . . . . .	9
2.2 Impact Based Fair Ranking . . . . .	10
2.2.1 Axioms for Fairness In Ranking . . . . .	10
2.2.2 Utility To Users . . . . .	10
2.2.3 Impact On items . . . . .	11
2.2.4 Exposure Based Fairness Violates Axioms . . . . .	11
2.2.5 Nash Social Welfare (NSW) policy . . . . .	13
2.3 Exposure Based Fairness . . . . .	14
2.3.1 Job Seeker Example . . . . .	14
2.3.2 Disproportionate CEO Example . . . . .	15
2.3.3 Diversity In Information Retrieval . . . . .	16
2.3.4 Fairness Constrained Ranking . . . . .	16
2.3.5 Probabilistic Ranking . . . . .	17

<b>3</b>	<b>Fairness In Ranking</b>	<b>20</b>
3.1	Learning To Rank . . . . .	22
3.2	Objective . . . . .	22
3.3	Learning to Rank as Policy Learning via ERM . . . . .	23
3.4	Fair Ranking policies . . . . .	23
3.5	Fairness Measures for Rankings . . . . .	24
3.5.1	Position Bias . . . . .	24
3.5.2	Exposure . . . . .	24
3.5.3	Exposure Allocation On Merit Basis . . . . .	25
3.5.4	Individual Fairness Disparity . . . . .	25
3.5.5	Group Fairness Disparity . . . . .	25
3.6	Plackett-Luce Ranking Policies . . . . .	26
3.7	Policy Gradient Training Algorithm . . . . .	27
3.8	Maximizing User Utility . . . . .	28
3.9	Minimising Disparity . . . . .	28
3.10	Our Contribution . . . . .	29
<b>4</b>	<b>Sampling Techniques</b>	<b>31</b>
4.1	Uniform Random Sampling . . . . .	31
4.1.1	How Uniform Sampling is done? . . . . .	32
4.2	Row Norm Sampling . . . . .	33
4.2.1	How Row Norm Sampling is Done? . . . . .	33
4.3	Leverage Score Sampling . . . . .	35
4.4	K-Medoid Sampling . . . . .	37
<b>5</b>	<b>Implementation and Dataset Details</b>	<b>39</b>
5.1	Dataset Details . . . . .	39
5.2	Implementations Details . . . . .	39
<b>6</b>	<b>Results</b>	<b>41</b>
6.1	Uniform Sampling Results . . . . .	42
6.1.1	NDCG Score . . . . .	42
6.1.2	DCG Score . . . . .	43
6.1.3	Average Ranking Measure . . . . .	44
6.1.4	ERR Score . . . . .	45
6.2	Row Norm Sampling Results . . . . .	46
6.2.1	NDCG Score . . . . .	46
6.2.2	DCG Score . . . . .	46

6.2.3	Average Ranking Measure . . . . .	47
6.2.4	ERR Score . . . . .	47
6.3	Leverage Score Sampling Results . . . . .	48
6.3.1	NDCG Score . . . . .	48
6.3.2	DCG Score . . . . .	48
6.3.3	Average Ranking Measure . . . . .	49
6.3.4	ERR Score . . . . .	49
6.4	K-Medoid Sampling Results . . . . .	50
6.4.1	NDCG Score . . . . .	50
6.4.2	DCG Score . . . . .	50
6.4.3	Average Ranking Measure . . . . .	51
6.4.4	ERR Score . . . . .	51
6.5	Time Stamp Result . . . . .	52
<b>7</b>	<b>Conclusion and Future work</b>	<b>53</b>
	<b>References</b>	<b>54</b>

# Abstract

Ranking is an important problem for a variety of applications. Classical algorithms for ranking may be unfair towards certain group of people or individuals. Fairness may be jeopardized by ranking algorithms that produce discriminatory results due to biased data or sampling methods. Hence in the past few years, algorithms to enforce fairness in ranking have been proposed. However they are computationally expensive. Hence it is better to train these on smaller samples of data. In this empirical study, multiple sampling strategies for fair ranking algorithms are compared and evaluated.

Uniform sampling, Leverage Score sampling, K -Medoid and Row Norm sampling are the four sampling strategies that are the subject of this study. The thesis tests and assesses the effectiveness of various sampling heuristics using a real-world data set i.e. Yahoo Learning To Rank Challenge Data set.

Our work shows that all heuristics perform reasonably well when compared with full data set, at the same time, giving impressive benefits in terms of computation time. It is an open question to obtain some theoretical guarantees for these sampling strategies for fair ranking algorithms.

# List of Figures

2.1	Example of Ranking[1]	9
2.2	Unfairness in Ranking Example[1]	9
2.3	An illustrative ranking problem example demonstrating the impact fairness and exposure allocations of various ranking policies	13
2.4	A classic Job seeking Example	15
2.5	Disproportionate number of male CEOs. Example	15
6.1	NDCG scores for Row Norm Samplings	46
6.2	DCG scores for Row Norm Samplings	46
6.3	Average Rank Measures for Row Norm Samplings	47
6.4	ERR scores for Row Norm Samplings	47
6.5	NDCG scores for Leverage Score Samplings	48
6.6	DCG scores for Leverage Score Samplings	48
6.7	Average Ranking Measures for Leverage Score Samplings	49
6.8	ERR scores for Leverage Score Samplings	49
6.9	NDCG scores for K-medoid Samplings	50
6.10	DCG scores for K-medoid Samplings	50
6.11	Average Ranking Measures for K-medoid Samplings	51
6.12	ERR scores for K-medoid Samplings	51
6.13	Time Stamps for Various Sample Sizes for Different Sampling Tech- niques	52



## CHAPTER 1

# Introduction

Ranking algorithms are vastly used in various domains. For e.g. search engines, recommendation systems, and e-commerce platforms to help users find the most relevant and useful information. However, the fairness of these algorithms has been a growing concern due to their potential to discriminate against certain groups of individuals. In recent years, the concept of fairness in ranking algorithms has received significant attention from academia and industry.

Adding additional constraints or objectives to the ranking process is one way to assure fairness in the algorithms used for ranking. However, this typically increases the computational time required to implement the algorithms. It could be beneficial to use sampling-based algorithms, which randomly choose a portion of items from the population. However, the particular sampling heuristics employed determines how effective the fairness of the algorithm is.

In today's multifaceted internet economies (such as online marketplaces, job searches, rental markets, and media streaming), interfaces based on rankings are common. It is well known that an item's ranking position has a significant impact on its exposure and financial success. In these systems, the things to be rated are products, job seekers, or other entities that convey economic benefits[2]. Unexpectedly, the algorithms that are employed to learn these rankings are frequently unaware of the impact they have on the products. There is evidence that the learning algorithms do not necessarily produce rankings that would be viewed as fair or desirable, but rather they aim to maximize the utility of the rankings to the users making queries to the systems[2].

Despite the expanding impact of online information systems on our culture and economy, fairness for rankings has received less attention than fairness in supervised learning for classification. Some of the work that has already been done

takes into account group fairness in rankings along the lines of demographic parity, proposing definitions and techniques that reduce the disparity in representation across groups in a prefix of the rating.

This thesis presents an empirical study of sampling heuristics for fair ranking algorithms. The goal of this study is to investigate the impact of different sampling heuristics on the fairness and performance of ranking algorithms. We mainly focus on a fair ranking algorithm called FAIR PG Rank.

In this study, we evaluate the performance of several sampling heuristics, including uniform sampling, Row Norm sampling, K - medoid sampling, and Leverage Score sampling. We apply these sampling heuristics to one fair ranking algorithm - the Fair-PG-Rank algorithm which is defined by a of Plackett-Luce ranking policy, and compare their performance in terms of various metrics like NDCG, DCG, Average Rank, and ERR .

The rest of this thesis is organized as follows. Chapter 2 provides an overview of related work in the field of fairness in ranking. Chapter 3 describes the methodology used in this study the fair PG rank algorithm given by [2]. The chapter describes the sampling strategies used. Chapter 5 gives the implementation details and Chapter 6 gives the results of the empirical study. Finally, Chapter 7 concludes the thesis and talks about future open directions for research in the field.

# Ranking

Ranking refers to the process of arranging a set of items or entities in a particular order based on their relevance, importance, or value. When provided with a collection of entities denoted as  $(i_1, i_2, \dots, i_N)$ , a ranking algorithm generates a ranking, represented by the assignment (mapping) of entities to specific ranking positions, denoted as 'r' [3].

The ranking process relies on an assessment of the entities' relative quality in relation to the specific task being performed.[3]. For example, the items returned by a search query are sorted mostly according to how relevant they are to the query. The utility will also be used to refer to the quality measure in the following. In general, a fair ranking is one in which the values of the protected attributes of the entities are not unjustifiably used to influence their placement [3].

Recommendation systems pull relevant content for users based on their histories and profiles. The historical data used in recommendation systems can consist of user ratings assigned to items or interactions with items, such as views or clicks, depending on the specific application [3]. For an item  $i$  and user  $u$ , recommenders typically estimate a rating,  $s(u,i)$ , which indicates the user  $u$ 's preference for the item  $i$ , or, in other words, the significance of item  $i$  for the user  $u$ . The items with the highest projected score for user  $u$  are then included in a suggestion list that is created for them. These ratings might be thought of as the recommenders' usefulness ratings[3].

In general, a recommendation is considered fair if the protected attributes of the user or item have no effect on the recommendation's outcome.

# Fairness in Machine Learning

Fairness in machine learning refers to the idea that the algorithms and models used to make decisions or predictions should not exhibit any biases or discrimination based on certain protected attributes such as race, gender, religion, age, or disability [4].

A fair machine learning model should treat all individuals fairly and equally, regardless of their background or characteristics. To ensure fairness in machine learning, it is important to identify and mitigate any potential sources of bias or discrimination in the data used to train the models. This may involve carefully selecting the variables and features used in the models, testing for and correcting any imbalances in the data, and monitoring the performance of the models to detect and address any disparities in outcomes[4].

Ultimately, the goal of fairness in machine learning is to create models that are transparent, accountable, and unbiased, and that promote equality and justice for all individuals.

Algorithmic bias is a well-researched and well-known issue in machine learning. Results may be biased by a variety of circumstances, and as a result, may be seen unfairly by some groups or people[4]. Personalized news delivery on social media platforms is one example.

Training data and the learning algorithm, the two key building blocks of a machine learning system, are also the major sources of biases that might provide unjust outcomes in machine learning tasks [4]. In light of this, biases are divided into two categories: Algorithmic and Data bias

## **1.1 Algorithmic Bias**

Unfairness may result from the biases inherent in algorithms, such as when certain optimization techniques are used improperly or biased estimators are used. Here, we give two illustrative algorithmic biases that may result in injustice and have an impact on user experience [5].

### **1.1.1 Presentation Bias**

When information is presented in biased ways, presentation bias occurs. The top-ranked results are typically thought of as the most relevant results by users and will therefore receive more clicks, whereas the lower-ranked results will receive less exposure even if they are also extremely relevant[5]. This is an example of ranking bias, which is a common problem in ranking problems.

### **1.1.2 Evaluation Bias**

The assessment bias often emerges when the wrong benchmarks are used to evaluate the model. Examples of standards that favor one gender or another when assessing facial recognition systems are Audience and IJB-A.

## **1.2 Data Bias**

The data itself contains the majority of the biases in machine learning. Data production, data collecting, and data storage methods can all introduce bias into the data.

### **1.2.1 Statistical Bias**

The process of collecting data and analysis often results in statistical bias. storage. It happens when errors are made in the design of the experiment or the gathering of the data, and as a result, the results are not an accurate picture of the population.

### **1.2.2 Pre-existing Bias**

The bias may already be present in the data throughout the generating process, even if the data are precisely sampled and chosen. Pre-existing bias can happen

when the data itself reflects biased judgments, which often renders the system no longer neutral and equitable[5].

### **1.2.3 Other Causes**

It is significant to note that there may be other reasons for unfairness besides prejudice, thus researchers should use vigilance at all times. The conflict between several fairness criteria is one instance. Because several fairness standards cannot be met concurrently, research has demonstrated that maintaining one form of fairness may prevent the violation of another.

## **1.3 Real World Example for Unfairness In Machine Learning**

Using a ranking algorithm in the criminal justice system is a real-world example of COMPAS (Correctional Offender Management Profiling for Alternative Sanctions)[4]. It is a tool that aids judges and parole boards in determining whether to sentence or release a criminal by forecasting their propensity to commit new crimes.

Though studies have proven that the COMPAS algorithm can provide biased results based on race, gender, and other factors, questions have been raised concerning its fairness. For instance, a 2016 ProPublica analysis discovered that COMPAS was twice as likely to incorrectly classify Black defendants as a high risk compared to White defendants and that it was also twice as likely to classify low-risk White defendants as High Risk compared to Black defendants[4].

Discussions about the function of algorithms in the criminal justice system and the significance of ensuring that these algorithms are impartial and fair have resulted as a result of this. The COMPAS example emphasizes the necessity for thorough assessment and monitoring of ranking algorithms in practical applications to make sure that they are not maintaining biases or inequities that already exist[4].

## 1.4 Why Fairness is necessary?

Fairness is necessary for machine learning for several reasons. First, it is a matter of ethics and social responsibility. Machine learning algorithms and models can have a significant impact on people's lives, from hiring decisions to loan approvals to criminal justice outcomes. If these models exhibit bias or discrimination, they can perpetuate and amplify existing inequalities and harm individuals and groups who are already marginalized or disadvantaged.

Second, fairness is essential for the accuracy and reliability of machine learning models. Biases in the data or model can lead to inaccurate predictions, misclassifications, or false positives, which can have significant consequences for individuals and society as a whole.

Finally, fairness is necessary to build trust in machine learning systems. If people do not believe that the models are fair and impartial, they may not use or trust them, which can limit the potential benefits that these systems can provide.

## CHAPTER 2

# Other Related Work

In this chapter, we describe some definitions and algorithms for fairness in ranking. All the examples and mathematical formulation are from different papers [6, 7] and others and are just reproduced here for completeness.

Fairness in ranking refers to making sure that there are no biases or discrimination present in the algorithms used to sort and display goods or people. To guide customers to the most relevant and helpful products or information, ranking algorithms are widely employed in a variety of applications, including search engines, e-commerce platforms, and recommendation systems[8].

Fairness has become a crucial factor in algorithmic decision-making. This is unfair when an agent with higher merit receives a worse result than an item with lower merit[9]. When a principal (or decision maker) must divide a finite resource among several entities, a generally acknowledged fairness principle states that an item should not receive more of a resource than item A if B does not have stronger merits for the resource than agent A[9]. Merit may be a requirement (such as job performance), a need (such as disaster relief), or another criterion of eligibility depending on the situation.

When making judgments, a principal or algorithm never has access to an item's genuine worth; instead, they rely on proxy attributes that can only partially predict merit (such as a student's GPA, star rating, or letter of recommendation)[9]. All of them fall short of accurately capturing an agent's merit, although prior methods have largely focused on defining fairness concepts based on actual features and results.

Overall, encouraging equality and reducing any harm resulting from biased or discriminatory ranking algorithms depend on fairness in ranking[8].



## 2.1 Unfairness in Ranking: An Example

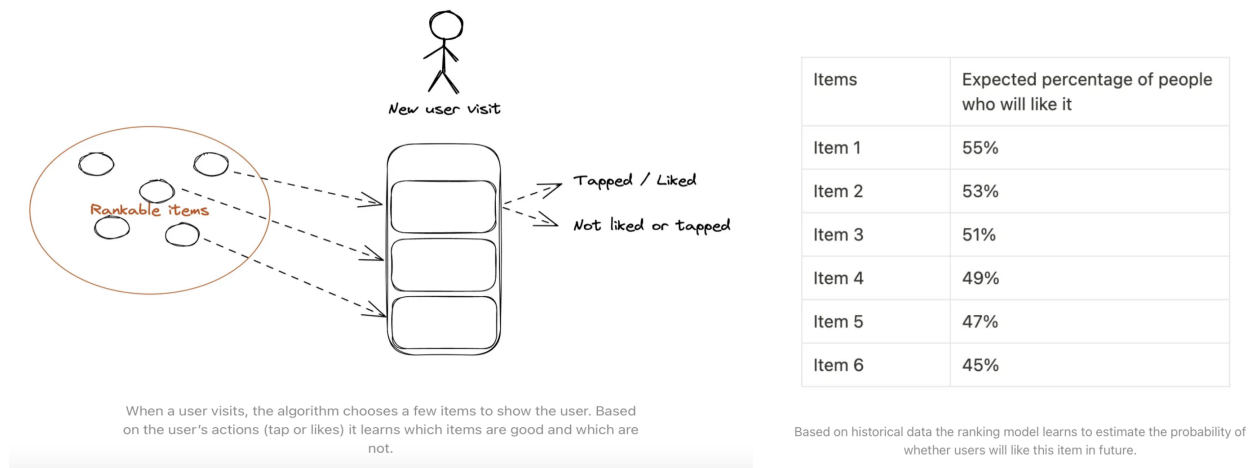


Figure 2.1: Example of Ranking[1]

### Example Of Biased Ranking

Items	Expected percentage of people who will like it	Percent times this item is at the top of list
<b>Item 1</b>	55%	<b>100%</b>
<b>Item 2</b>	53%	0%
<b>Item 3</b>	51%	0%
<b>Item 4</b>	49%	0%
<b>Item 5</b>	47%	0%
<b>Item 6</b>	45%	0%

Although the number of users who like purple are just slightly less than green, 100% of the time the item on top is green!

Figure 2.2: Unfairness in Ranking Example[1]

## 2.2 Impact Based Fair Ranking

The entire section relies heavily on [7] and all proofs and examples are from the paper. We first describe the Impact Based Fairness in Ranking. Fairness in ranking may be thought of as an issue of resource distribution, where the resources are the places in a ranking that must be distributed appropriately among the objects. Here we describe fairness axioms that complement widely acknowledged ideas from the fair-division literature in order to succinctly describe fairness:

### 2.2.1 Axioms for Fairness In Ranking

- A resource allocation is **envy-free** if for every pair of individuals  $i$  and  $j$ ,  $i$  does not prefer  $j$ 's allocation more than their own. In other words, no individual should feel envious or desire someone else's allocation more than their own[7].
- The second axiom, "Dominance over Uniform Ranking" focuses on ensuring that the ranking produced by an algorithm is not dominated by a uniform ranking, which provides equal scores or ranks to all individuals[7].
- The Pareto Optimality Principle: Pareto optimality represents an allocation of resources or outcomes where it is impossible to improve the well-being of one individual without reducing the well-being of another individual[7].

### 2.2.2 Utility To Users

The utility is the usefulness of the item to the user. So through a utility function of the following design, we gauge the usefulness a policy  $\pi$  offers to its consumers.

$$U(\pi) := \sum_{u \in \mathcal{U}} \sum_{i \in \mathcal{I}} \sum_{k=1}^n e(k) r(u, i) X_{u,i,k}^{\pi}$$

where  $X^{\pi}$  is the tensor whose  $(u,i,k)$  element

$$X_{u,i,k}^{\pi} := P(\sigma(i) = k | \pi, u)$$

is marginal probability of item  $i$  being ranked at the  $k$ -th position for user  $u$  under policy  $\pi$ , and  $(X_u^\pi, *, *)$  should be the doubly stochastic matrix [7]. By utilizing this approach, it is possible to express the impact of policy  $\pi$  with fewer parameters. Since all stochastic ranking strategies with the same matrix have the same user utility, we specifically employ just  $I^2$  parameters for user  $u$  instead of the exponential number of potential rankings[7].

### 2.2.3 Impact On items

Although rating things according to their likelihood of relevance increases user value, several previous works have emphasized that this naive approach might provide rankings that are unjust to the objects. It specifically proposes a notion of fairness mandating that amortized exposure should be allocated proportionally to their amortized merit and measures similarity between individual things by their amortized merit[7]. This notion of fairness strives to distribute exposure equally among items of comparable worth. This formulation, however, lacks a compelling rationale for why exposure should be related proportionally to relevance or linked via any other particular function, as we have already demonstrated[7]. As we know, Exposure is merely a secondary concern for the products; instead, they are primarily concerned with how the ranking would affect them. Therefore we will focus on Fairness of Impact, where impact measures the influence a ranking policy has on a certain item  $i$ . So we will focus on Item centric Impact on matrix  $X_{*,i,*}$ . The Impact on each Item will be,

$$Imp_i(X_{*,i,*}^\pi) := \sum_{u \in \mathcal{U}} \sum_{k=1}^n v_i(u, k) X_{u,i,k}^\pi$$

Where,  $v_i(u, k)$  is an Impact function that specifies the amount of impact item  $i$  has for user  $u$  when it is ranked at position  $k$ . Also, the utility of the user is equal to the summation of Impacts on  $i \in I$ , a set of items.

### 2.2.4 Exposure Based Fairness Violates Axioms

In literature, it has been demonstrated in detail how the traditional fairness of exposure paradigm may foster "envy" among the items and provide an unjust

effect distribution in light of our axioms, demonstrating the necessity for new algorithms to ensure fairness of impact [7]. Exposure must be allocated proportionally to amortized merit under exposure-based fairness. So following modified optimization was proposed in [7], which uses a more broad constraint in which merit is measured using an application-dependent link function  $f(\cdot) > 0$ .

$$\begin{aligned}
& \pi_{\text{expo-fair}} \\
& = \arg \max_{\{X_{*,i,*}^\pi\}_{i \in I}} \sum_{u \in \mathcal{U}} \sum_{i \in I} \sum_{k=1}^n e(k)r(u, i)X_{u,i,k}^\pi \left( = \sum_{i \in I} \text{Imp}_i(X_{*,i,*}^\pi) \right), \\
& \text{s.t. } \frac{\text{Exp}_i(X^\pi)}{f(\text{Merit}_i)} = \frac{\text{Exp}_j(X^\pi)}{f(\text{Merit}_j)}, \quad \forall i, j \in I, \\
& \quad \sum_{k=1}^n X_{u,i,k}^\pi = 1, \quad \forall (u, i) \\
& \quad \sum_{i \in I} X_{u,i,k}^\pi = 1, \quad \forall (u, k) \\
& \quad 0 \leq X_{u,i,k}^\pi \leq 1, \quad \forall (u, i, k)
\end{aligned}$$

From the above formulae, we can clearly observe that if we take the "arg max " of the Utility of the user then it will give us the Exposure based Fair Ranking on the basis of policy  $\pi$  [7] where  $\text{Merit}_i$  is the relevance of item  $i$  over all users and  $\text{Exp}_i(X^\pi)$  is the amount of exposure for item  $i$  under policy  $\pi$ . For proving that Exposure based fairness violates axioms we will take a counterexample by referring to the table above in Fig.3. This example and table is taken from [7]. So let's assume two users  $u_1$  and  $u_2$  and item  $i_1$  and  $i_2$ . Also, we'll assume that the item which is ranked at the top will get exposed to users. So  $e(1)=1$  and  $e(2)=0$ . We'll focus on three ranking policies given in the above table in fig.3. which are utility maximizing, Exposure Based, and Uniform Random Policy[7].

So from the above table, we can observe that the three policies' exposure distribution and the degree of influence on each item are shown in Tables in Fig. (2.3). All exposure is given to the item( $i_1$ ) with the highest significance by  $\Pi_{\text{Max}}$ . This results in the unjust circumstance where  $i_2$  receives no exposure while having a significant relevance and the maximum user utility ( $U \pi(\text{max}) = 1.3$ ).  $\pi_{\text{Expo-fair}}$  implements the exposure limitation to address this unjust distribution. [7]In particular, it distributes some exposure to the item ( $i_2$ ) that is less important and makes



$$\begin{aligned}
\pi_{NSW} = \arg \max_{\{X_{*,i,*}^\pi\}_{i \in \mathcal{I}}} & \prod_{i \in \mathcal{I}} Imp_i(X_{*,i,*}^\pi), \\
\text{s.t.} & \sum_{k=1}^n X_{u,i,k}^\pi = 1, \quad \forall(u, i) \\
& \sum_{i \in \mathcal{I}} X_{u,i,k}^\pi = 1, \quad \forall(u, k) \\
& 0 \leq X_{u,i,k}^\pi \leq 1, \quad \forall(u, i, k)
\end{aligned}$$

single item with zero impact, the NSW is zero[7]. In contrast, the traditional goal of maximizing user utility which equals the aggregate of impacts can be achieved even if certain elements have no influence at all[7]. Empirical results in literature have shown that  $\pi_{NSW}$  is (almost) envy-free and Pareto optimum.

## 2.3 Exposure Based Fairness

To better understand Exposure Based Fairness we present two examples from literature that show how Biased Allocation of Opportunities can create unfairness among the group of people or individuals themselves[6].

### 2.3.1 Job Seeker Example

Think about a website that links employers (users) with job candidates (items). The example that follows shows how even slight changes in item relevance can result in significant disparities in exposure and, consequently, in economic opportunity between groups. In this instance, the web service employs a ranking-based approach to show a group of 6 candidates to pertinent firms for a software engineering post (Figure 2.4). Three men and three women make up the set. For the employers, the relevance of the male candidates is 0.80, 0.79, and 0.78, respectively, whereas it is 0.77, 0.76, and 0.75 for the female applicants. Following the conventional probabilistic definition of relevance, in this case, 0.77 denotes that

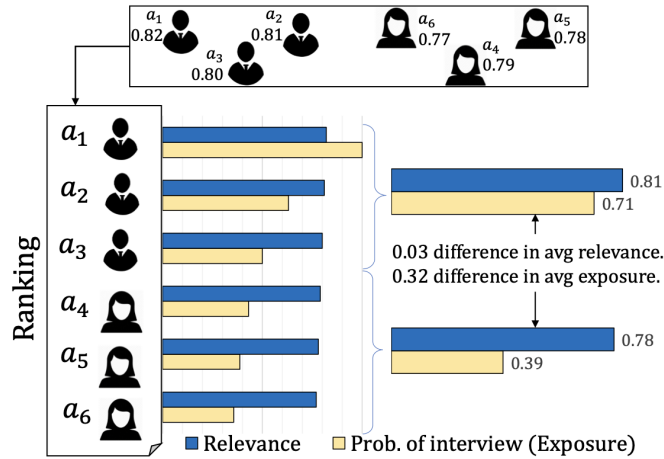


Figure 2.4: A classic Job seeking Example

77% of the employers who are the subject of the inquiry consider the applicant to be relevant. According to the Probability Ranking Principle, these candidates should be ranked in decreasing order of significance, with the three male candidates at the top, then the female candidates. What does this signify in terms of the two groups' exposure? Even if there is only an average difference in relevance between male and female applications of 0.03, female applicants would receive 30% less exposure if we adopt the typical exposure drop-off (i.e., position bias) of  $1/\log(1 + j)$ , where  $j$  is the position in the ranking[6], it seems appropriate to share exposure more equitably[6].

### 2.3.2 Disproportionate CEO Example



Figure 2.5: Disproportionate number of male CEOs. Example

Fairly depicting the results' distribution. Sometimes, either expressly or implicitly, the outcomes of a query are utilized as a statistical sample. For instance, a

user would anticipate that a "CEO" search on a search engine will yield about the proper number of executives, both male and female, representing the actual distribution of male and female CEOs in the globe. A search engine may be seen as prejudiced if it returns a much higher proportion of males than females, as in the hypothetical results in the above Figure(2.5). In fact, the research found that gender bias was present in picture search results for a range of professions[6]. It has been demonstrated that these biases do in fact influence people's beliefs about different jobs. A biased information environment may have an impact on users' perceptions and behavior. This means that the best ranking according to the Probability Ranking Principle may still appear in that way even if users' relevance distribution matches the actual distribution of female CEOs. Even though there may be a decrease in utility for the consumers, it appears appropriate to distribute exposure according to relevance rather than just depending on the PRP[6].

### **2.3.3 Diversity In Information Retrieval**

Since they both produce more diverse ranks, fairness and variety in rankings may initially appear to be mutually exclusive. However, their driving forces and operating systems are essentially unlike. Diversified ranking, like the PRP, is only focused on enhancing user utility, but this approach[6] to fairness strikes a balance between users' and goods' demands. Particularly, only the utility measure that is maximized differs between the PRP and diversified ranking; both maximize benefit for the user alone. The utility metric takes into account uncertainty and declining rewards from numerous relevant outcomes under extrinsic diversity. The utility metric, which falls under intrinsic diversity, treats ranks like portfolios and takes redundancy into account[6]. Additionally, the goal of exploration variety is to increase user utility over time by facilitating more efficient learning. The work on fairness in this paper[6] is significantly different in terms of its goal and technique since it adds rights to the objects that are being rated rather than changing the utility measure for the user.

### **2.3.4 Fairness Constrained Ranking**

Recognizing the prevalence of rankings across applications, [6] hypothesizes that fairness is context- and application-dependent and that there is no universal description of what makes a fair rating. For example, shown below that various concepts of fairness entail various utility trade-offs, which may be appropriate in one circumstance but not in another. This part establishes a framework for artic-



ulating fairness requirements on ranks and then computes the utility-maximizing ranking according to these fairness constraints with verifiable guarantees in order to handle this spectrum of potential fairness restrictions[6]. For the sake of simplicity, let's assume that there is just one query,  $q$ , and that we wish to offer a ranking "r", of a collection of documents,  $D = (d_1, d_2, d_3, \dots, d_N)$ . The problem of optimum ranking under fairness constraints may be expressed as the following optimization problem by denoting the utility of a ranking  $r$  for query  $q$  with  $U(r|q)$ [6]:

$$r = \operatorname{argmax}_r U(r|q),$$

$r$  is fair. In this approach, the Probabilistic Ranking Principle's objective which manifests as the particular case of no fairness constraints is generalized. The following four elements will be specified in order to properly instantiate and solve this optimization challenge. They begin by defining a broad category of utility measures,  $U(r|q)$ , which includes a number of widely used ranking metrics. Second, they expand the class of rankings to probabilistic rankings in order to address the issue of how to optimize over rankings, which are discrete combinatorial objects[6]. Thirdly, they rephrase the optimization issue as an effectively solvable linear program, which entails the need for a practical yet expressive language to convey fairness restrictions. Finally, they demonstrate how to effectively extract a probabilistic ranking from the linear program's answer.

### 2.3.5 Probabilistic Ranking

$$\begin{aligned} U(R|q) &= \sum_r R(r) \sum_{u \in \mathcal{U}} P(u|q) \sum_{d \in \mathcal{D}} v(\operatorname{rank}(d|r)) \lambda(\operatorname{rel}(d|u, q)) \\ &= \sum_r R(r) \sum_{d \in \mathcal{D}} v(\operatorname{rank}(d|r)) u(d|q) \end{aligned}$$

Ranks are combinatorial objects, therefore it would take exponentially more time in  $|D|$  to naively search the space of all rankings for a utility-maximizing ranking under fairness restrictions[6]. Instead of using a single deterministic ranking  $r$ , they examine probabilistic rankings  $R$  in order to prevent such combinatorial optimization[6]. So they readily applied the idea of utility to probabilistic rankings since a probabilistic ranking ( $R$ ) is a distribution across rankings[6]. where notations are:

- $u$  = user
- $q$  = query entered by the user.
- $d$  = document returned for the particular fired query.
- $\lambda, v$  = These two are application dependent functions.
- $v(rank(d|r))$  = It models how much attention document  $d$  gets at rank  $rank(d|r)$ .
- $\lambda(rel(d|u, q))$  = It maps relevance of document for a user to its utility[6].

Let  $P_{i,j}$  be the probability that  $R$  places document  $d_i$  at rank  $j$ , so  $P$  will be a "doubly stochastic" matrix [6]. Which means,  $\sum_i P_{i,j} = 1$  and  $\sum_j P_{i,j} = 1$ . So the Probabilistic Ranking Utility will be,

$$U(P|q) = \sum_{d_i \in \mathcal{D}} \sum_{j=1}^N P_{i,j} u(d_i|q) v(j).$$

To make the notation more simple, we can write "utility of the ranking as a matrix product". So we will take two vectors " $u$ " and " $v$ ", where, //

- $u_i = u(d_i|q)$
- $v_j = v(j)$

So, the Utility now can be written as,

$$U(P|q) = \mathbf{u}^T \mathbf{P} \mathbf{v}$$

In sections (2.2) and (2.3) we discussed Impact based fairness and Exposure based fairness respectively. In Impact-based fairness we observed that it was unable to satisfy the fairness axioms so we went for Exposure-based fairness which proved to somehow satisfies the fairness axioms. But in Exposure based fairness the exposure allocated to the items is an accidental by-product in order to blindly maximize the utility to users which will ultimately be responsible for creating an endogenous bias. Instead of this, the allocation of exposure should be based on merit specified explicitly i.e. the exposure should be allocated explicitly. So for this purpose, we further went for the Fair-PG-Rank[2] as it extends the work of Exposure-based Fairness[6] to overcome the above-mentioned problem of explicitly exposure allocation.

## CHAPTER 3

# Fairness In Ranking

Fairness in ranking can be defined as below mentioned points in a simplified way.

- Fairness in ranking refers to ensuring that the ranking algorithm does not display any biases or discrimination that could cause certain items or people to be unfairly disadvantaged or favored.
- In order to minimize the possible harm that can result from biased or discriminatory outcomes, particularly in terms of exposure and visibility, fairness in ranking entails constructing and optimizing ranking algorithms.
- By guaranteeing that comparable item individuals are ranked equally across the board, regardless of their origins or individual traits, and that various populations or groups are fairly represented in the ranking algorithm, fairness in ranking seeks to advance equality.
- The qualities and criteria that are used to rank the items or the individuals must be carefully chosen, any data imbalances must be tested for and corrected, and the performance of the algorithms must be watched to identify and address any discrepancies in results.
- Individual fairness, group fairness, and algorithmic transparency are three ways to ensure that rankings are fair. These three methods also involve making sure that the ranking algorithm is transparent to users and does not unfairly disadvantage any particular group or population.

Overall, as ranking algorithms are widely employed in many applications, including search engines, e-commerce platforms, and recommendation systems, fairness in ranking is a critical topic in machine learning and artificial intelligence. We can lessen the potential harm that could result from biased or discriminatory outcomes and make sure that these algorithms function for the advantage of all users by encouraging fairness and equality in them.

# Fair PG Rank

Traditional Learning-to-Rank (LTR) techniques maximize the ranking's usefulness to users, but they are unaware of how their actions affect the rated items. However, there is growing recognition that the latter must be taken into account for a variety of ranking applications. (e.g. online marketplaces, job placement, admissions). In order to fill this gap, This paper[2] offers a generic LTR framework that may optimize a variety of utility measures (such NDCG) while adhering to the fairness of exposure restrictions with regard to the items.

This framework[2] adds stochastic ranking policies to the domain of learnable ranking functions, creating a language for precisely expressing fairness requirements. Additionally, using a policy-gradient approach, they offer a new LTR algorithm dubbed Fair-PG-Rank for directly exploring the space of fair ranking policies. They give empirical results on simulated and real-world data sets demonstrating the effectiveness of the technique in individual and group-fairness contexts, in addition to the theoretical evidence used to derive the framework and the algorithm[2].

Despite the expanding impact of online information systems on our society and economy, the topic of ranking fairness has received comparatively little attention. Some people view group fairness in rankings as being similar to demographic parity, providing definitions and techniques that reduce the disparity in representation between groups in the prefix of the rating.

Recent studies like[6] have made the case that the fairness of ranking systems is related to the way they distribute exposure to distinct items or groups of related items based on merit. The fairness requirements that are explicitly linked between relevance and exposure in expectation or amortized over a group of inquiries are specified and enforced in these studies[2]. So LTR(Learning-To-rank) algorithm comes into the picture in order to overcome and solve this problem of explicit exposure allocation.

## 3.1 Learning To Rank

The initial Learning-to-Rank (LTR) algorithm[2], called Fair-PG-Rank, focuses not only on optimizing utility for users but also on enforcing strict limitations on merit-based exposure for items. These fairness constraints are essential for compliance with anti-trust laws, addressing winner-takes-all dynamics in music streaming services, implementing anti-discrimination measures, or ensuring variations of search neutrality. The algorithm[2] places emphasis on fairness in relation to the allocation of exposure based on the merit of items, and it incorporates fairness considerations into the learning process to identify biases in representation. This information is based on the research findings in this paper[2].

## 3.2 Objective

The main objective is to develop ranking policies where the distribution of exposure to items is not an unintended side effect of optimizing user utility, but rather where a merit-based exposure-allocation restriction may be specified and is imposed by the learning algorithm[2]. In a classic example, ranking 10 job candidates, where the probabilities of relevance (i.e., the likelihood that an employer will invite a candidate for an interview) are respectively (0.89, 0.89, 0.89, 0.89) for 5 male candidates and (0.88, 0.88, 0.88, 0.88) for 5 female candidates. If the consumers' utility were maximized for nearly all information retrieval measures, these 10 choices would be sorted by likelihood of relevance[2].

Although having about the same significance, the female candidates (ranked 6,7,8,9,10) would receive far less attention than the male candidates (ranked 1,2,3,4,5). Hence, a large endogenous bias against female candidates is produced by the ranking mechanism itself, greatly increasing any external prejudice that the employers may have. Regarding the endogenous bias brought on by the system itself, we contend that exposure allocation should be able to be explicitly specified (e.g., make exposure proportional to relevance) and that the ranking policy accurately learns this specified exposure allocation[2]. They develop their fair LTR framework as guided by the following three goals:

*Goal 1:* Exposure allocated to an item is based on its merit. More merit means more exposure.

*Goal 2:* Enable the explicit statement of how exposure is allocated relative to the merit of the items.

*Goal 3:* Optimize the utility of the rankings to the users while satisfying *Goal 1* and *Goal 2*.

In order to further refine these goals they first formulate LTR in context with ERM i.e. "Empirical Risk Minimization" in the further upcoming section.

### 3.3 Learning to Rank as Policy Learning via ERM

Let  $Q$  be the distribution from which queries are drawn.

- $d^q = (d_1^q, d_2^q, \dots, d_n^q)$ , where  $d^q$  is the candidate set of the documents related to  $q$  [2]. This candidate set has to be ranked.
- $r^q = (r_1^q, r_2^q, \dots, r_n^q)$ , where  $r^q$  is the relevance corresponding to the above candidate set document [2].
- $x_i^q = \Psi(q, d_i^q)$ , where  $x_i^q$  is a feature vector that explains the match between  $d_i^q$  and  $q$  [2].

Now finding a ranking policy  $\pi^*$  that maximizes the anticipated utility of  $\pi$  is the standard objective in Learning To Rank.

$$\pi^* = \operatorname{argmax}_{\pi \in \Pi} \mathbb{E}_{q \sim Q} [U(\pi|q)]$$

where  $U(\pi|q)$  is the expectation of a ranking metric  $\Delta$  over  $\pi$ , given below

$$U(\pi|q) = \mathbb{E}_{r \sim \pi(r|q)} [\Delta(r, \operatorname{rel}^q)]$$

Common choices for the  $\Delta$  are DCG, NDCG, Average Rank, and ERR.

### 3.4 Fair Ranking policies

Like in traditional LTR algorithms, They introduce a constraint into the learning problem that imposes an application-dependent idea of fair distribution of exposure rather than single-mindedly maximizing this utility metric.

They now describe the goal of fair LTR by limiting the set of acceptable ranking strategies to those that have anticipated disparity smaller than some parameter  $\delta$  [2]. Where  $D(\pi|q) \geq 0$  is a measure of Unfairness or the Disparity.

Using Lagrange's Multiplier and avoiding minimization w.r.t  $\lambda$  for a chosen  $\delta$ . Instead, choose a specific  $\lambda$  and then compute the corresponding utility/fairness trade-off[2]. This indicates that all that need to do is to find

$$\hat{\pi}_\lambda^* = \operatorname{argmax}_\pi \frac{1}{N} \sum_{q=1}^N U(\pi|q) - \lambda \frac{1}{N} \sum_{q=1}^N \mathcal{D}(\pi|q)$$

and then recover  $\delta_\lambda = \frac{1}{N} \sum_{q=1}^N D(\hat{\pi}_\lambda^*|q)$  afterwards.

So the third goal is implemented by this formulation but the concrete definition of "D" Disparity is still not clear which is covered in further sections of Group Fairness Disparity and Individual Fairness Disparity.

## 3.5 Fairness Measures for Rankings

In order to define Disparity or unfairness "D" clearly, a class of fairness measures for ranking is discussed in the paper[6] "Fairness In Ranking". These are Position Bias, Exposure, Merit-Based Exposure Allocation, Individual Fairness Disparity, and Group Fairness Disparity.

### 3.5.1 Position Bias

The percentage of users who visit a ranking and look at the item at position  $j$  is known as the position bias of position  $j$ , or  $v_j$ . Higher ranks are anticipated to gain more attention than lower ones, capturing the amount of attention that a result will garner[2].

### 3.5.2 Exposure

The expected attention that a document receives is referred to as its exposure[6]. This is the same as the anticipated position bias from every position the document might be placed. Exposure is denoted as  $v_\pi(d_i)$  and defined as ,

$$\text{Exposure}(d_i|\pi) = v_\pi(d_i) = \mathbb{E}_{r \sim \pi(r|q)} [\mathbf{v}_r(d_i)]$$

where  $r(d_i)$  = position of document  $d_i$  under ranking  $r$ .



### 3.5.3 Exposure Allocation On Merit Basis

The first two goals were that exposure should be based on an application-dependent notion of merit.  $M(rel_i) \geq 0 = \text{Merit of document } d_i$ . They assert that any candidate document should receive exposure proportional to its merit  $M_i$ .

$$\forall d_i \in d^q : \text{Exposure}(d_i|\pi) \propto M(rel_i)$$

To overcome the problem of overabundance of Exposure, which is sometimes due to very small merit i.e.  $\epsilon$  of some documents, the allocation of exposure is more than they deserve. This is called overabundance of Exposure[2]. So, They considered a particular inequality constraint.

$$\frac{\text{Exposure}(d_i|\pi)}{M(rel_i)} \leq \frac{\text{Exposure}(d_j|\pi)}{M(rel_j)}$$

where  $\forall d_i, d_j \in d^q$  and  $M(rel_i) \leq M(rel_j) > 0$

### 3.5.4 Individual Fairness Disparity

The following disparity measure D for the individual notion of fairness, which measures the extent to which the fairness-of-exposure restrictions are violated, may now be defined[2].

$$\mathcal{D}_{\text{ind}}(\pi|q) = \frac{1}{|H_q|} \sum_{(i,j) \in H_q} \max \left[ 0, \frac{v_\pi(d_i)}{M_i} - \frac{v_\pi(d_j)}{M_j} \right]$$

"where  $H_q = (i, j)$  s.t.  $M_i \geq M_j > 0$ . The measure  $D_{\text{ind}}(\pi|q)$  is always non-negative and it equals zero only when the individual constraints are exactly satisfied"[2].

### 3.5.5 Group Fairness Disparity

While other applications call for a group-based notion of fairness, the disparity measure from above implements an individual notion of fairness. In this case, fairness is summed up among each group's members.

A collection of papers may refer to bundles of items offered for sale by a single vendor in an online marketplace, to articles written by a single author, or to job applicants who identify as members of a protected class.

They provide exposure to groups proportionate to their merit, much like in the case of individual fairness. So they define the group fairness disparity for query (q) as follows in the case of just two groups,  $G_0$  and  $G_1$ .

$$\mathcal{D}_{\text{group}}(\pi|q) = \max\left(0, \frac{v_{\pi}(G_i)}{M_{G_i}} - \frac{v_{\pi}(G_j)}{M_{G_j}}\right)$$

"where  $G_i$  and  $G_j$  are such that  $M_{G_i} \geq M_{G_j}$  and  $\text{Exposure}(G|\pi) = v_{\pi}(G) = \frac{1}{|G|} \sum_{d_i \in G} v_{\pi}(d_i)$  is the average exposure of group  $G$ , and the merit of the group  $G$  is denoted by  $M_G = \frac{1}{|G|} \sum_{d_i \in G} M_i$ "[2].

### 3.6 Plackett-Luce Ranking Policies

In the previous section, a generic framework for learning ranking rules under fairness-of-exposure restrictions is established. What has to be demonstrated is that given the disparities  $D$  described above, there exists a stochastic policy class and an associated training method that can achieve the objective in Equation given in section(3.4). In order to introduce Fair-PG-Rank, Firstly "Plackett-Luce Ranking Policies" has to be described specifically and then effectively optimize training objective by a policy gradient method.

The following definitions of ranking policies  $\pi$  have two parts: a scoring model that specifies a distribution over rankings and the corresponding sampling technique. By permitting any differentiable machine learning model with parameters  $\theta$  starting with the scoring models  $h_{\theta}$ , such as a linear model or a neural network.  $x^q$  is the input showing feature vectors of all query-document pairs of the candidate set[2].  $h_{\theta}(x^q) = (h_{\theta}(x_1^q), h_{\theta}(x_2^q), h_{\theta}(x_3^q), \dots, h_{\theta}(x_n^q))$  is a vector of scores output by the scoring model, so based on these score vectors the probability  $\pi_{\theta}(r|q)$  of a ranking  $r = \langle r(1), r(2), \dots, r(n_q) \rangle$  under the Plackett-Luce model is the following product of softmax distributions[2].

$$\pi_{\theta}(r|q) = \prod_{i=1}^{n_q} \frac{\exp(h_{\theta}(x_{r(i)}^q))}{\exp(h_{\theta}(x_{r(i)}^q)) + \dots + \exp(h_{\theta}(x_{r(n_q)}^q))}$$

the derivative of  $\pi_{\theta}(r|q)$  and  $\log\pi_{\theta}(r|q)$  exists whenever the scoring model  $h_{\theta}$  is differentiable. It is also effective to sample a ranking using the Plackett-Luce model[2].

The next step is looking through this policy  $\Pi$  space for a model that maximizes the goal in the aforementioned equation. So in the next section, a policy gradient algorithm is proposed.

### 3.7 Policy Gradient Training Algorithm

This section[2] suggests a policy-gradient strategy that repeatedly enhances the ranking policy using stochastic gradient descent (SGD) updates. Since  $U$  and  $D$  are expectations over rankings sampled from, computing the gradient by brute force is not possible.

In this section[2], the required gradients over expectations are obtained as an expectation over gradients. We then estimate this expectation as an average across a small sample of ranks from the policy to provide a rough gradient.

In order to maximize user utility, traditional LTR approaches either use heuristics to optimize over probabilistic formulations of rankings or are designed to optimize over a smoothed version of a certain utility measure, such as SVMRank, RankNet, etc[2].

While the LTR setup is similar to ListNet, they directly optimize over utility and disparity metrics using a policy gradient technique rather than a heuristic loss function. In contrast to most traditional LTR methods, which optimize upper limits or heuristic proxy measures, policy-gradient learning directly optimizes the ranking policy[2].

First off, there are no limitations on the information retrieval (IR) metric that may be used because the learning system directly optimizes a given user utility mea-

sure. Second, since disparity measure D is likewise an expectation over ranks, The application of the same policy-gradient method to it as well is possible. Overall, the non-smoothness of ranks is simply handled by the application of policy-gradient optimization in the space of stochastic ranking policies[2].

So the main objective to achieve Fairness In Ranking is to ultimately Minimise the Unfairness or Disparity "D" and Maximising the Utility/Usefulness. In the next sections, the algorithm for minimizing the disparity and maximizing the utility is given by this paper[2].

### 3.8 Maximizing User Utility

Given that the space of ranks has an exponential cardinality, determining the gradient with respect to  $\theta$  this expectation is not an easy task. This is remedied by sampling using the log-derivative technique pioneered by the REINFORCE algorithm[2].

$$\nabla_{\theta} U(\pi_{\theta}|q) = \nabla_{\theta} \mathbb{E}_{r \sim \pi_{\theta}(r|q)} \Delta(r, \text{rel}^q) = \mathbb{E}_{r \sim \pi_{\theta}(r|q)} [\nabla_{\theta} \log \pi_{\theta}(r|q) \Delta(r, \text{rel}^q)]$$

"The above transformation exploits that the gradient of the expected value of the metric  $\delta$  over rankings sampled from  $\pi$  can be expressed as the expectation of the gradient of the log probability of each sampled ranking multiplied by the metric value of that ranking".

### 3.9 Minimising Disparity

Calculation of the gradient of the fairness-of-exposure term D when it is part of the training goal. Thankfully, it shares a structure with the utility term, making the Monte-Carlo method applicable so for the individual-fairness disparity measure, in particular, the gradient can be computed as[2]:

$$\nabla_{\theta} \mathcal{D}_{\text{ind}} = \frac{1}{|H|} \sum_{(i,j) \in H} \mathbb{1} \left[ \left( \frac{v_{\pi}(d_i)}{M_i} - \frac{v_{\pi}(d_j)}{M_j} \right) > 0 \right] \times \mathbb{E}_{r \sim \pi_{\theta}(r|q)} \left[ \left( \frac{v_r(d_i)}{M_i} - \frac{v_r(d_j)}{M_j} \right) \nabla_{\theta} \log \pi_{\theta}(r|q) \right]$$

( $H = \{(i, j) \text{ s.t. } M_i \geq M_j\}$ )

And for Group Fairness Disparity, the gradient can be derived as

$$\nabla_{\theta} \mathcal{D}_{\text{group}}(\pi|G_0, G_1, q) = \nabla_{\theta} \max(0, \xi_q \text{diff}(\pi|q)) = \mathbb{1}[\xi_q \text{diff}(\pi|q) > 0] \xi_q \nabla_{\theta} \text{diff}(\pi|q)$$

where  $\text{diff}(\pi|q) = \left( \frac{v_{\pi}(G_0)}{M_{G_0}} - \frac{v_{\pi}(G_1)}{M_{G_1}} \right)$ , and  $\xi_q = \text{sign}(M_{G_0} - M_{G_1})$ .

"Another important point is that in both the case of individual disparity and group disparity, the expectation can be estimated as an average over a sample of ranking from a distribution".

Above are the two algorithms given by [2] for the minimization of Individual Disparity and Group Disparity respectively.

### 3.10 Our Contribution

Impact-based fairness[7] and Exposure based fairness[6] have certain issues as discussed previously. So we selected and implemented the Fair-PG-Rank[2] as it overcomes the problems that arise in the previous two methods mentioned above.

We implemented the "Fair-PG-Rank" algorithm given in the paper[2] and observed that the time taken to train the model for the large (19,944) row data of the Yahoo Learning To Rank Challenge dataset is very large i.e. more than 13 hours . So we chose to go for the sampling techniques in order to check and observe if we can perform sampling on the above-mentioned large Yahoo dataset and take appropriate samples by applying some sampling techniques and can get desired near-about results in a comparatively small time as compared to a large full dataset.

So our main contribution in this thesis is to conduct an empirical study of sampling heuristics for the fair ranking algorithm. We aim to explore the performance of different sampling techniques in terms of their effectiveness in achieving fairness in the fair PG rank algorithm while maintaining accuracy and efficiency. Specifically, we will evaluate the performance of several sampling heuristics, including uniform sampling, Row Norm sampling, Leverage Score sampling, and K-Medoid sampling, on a real-world dataset.

Our contribution provides valuable insights into how various sampling techniques perform for Fairness based ranking algorithms.

So in the further sections firstly we briefly discussed the four sampling techniques which are Uniform Random Sampling, Row Norm Sampling, Leverage Score Sampling, and K-Medoid Sampling which we used in our research work, and then discussed the implementation and dataset information.

## CHAPTER 4

# Sampling Techniques

### 4.1 Uniform Random Sampling

Uniform sampling ensures that each data point or data of the population has an equal probability of being included in the sample, thereby minimizing bias and providing a representative sample for analysis[10].

In this sampling technique, a random number generator is typically employed to assign a random value to each individual or item in the population. These random values are then sorted, and the top N values are selected to form the sample, where N represents the desired sample size[11]. This process guarantees that every individual or item has an equal probability of being selected, as the random values distribute uniformly across the population[10]. This sampling approach allows all possible combinations of n units to be produced from the population of n units with the same chance of selection.

Ultimately this sampling ensures randomness and eliminating biases in order to obtain reliable and accurate results, it also allows for generalization of findings from the sample to the larger population, making it a fundamental tool in statistical inference[11]. So that this technique serves as a cornerstone in statistical analysis, providing a fair and unbiased approach to selecting samples from populations of interest.

By using this sampling technique, one can reduce the potential for selection bias and improve the reliability and validity of their work and findings. It also provides a fair and unbiased approach to selecting samples, enabling more accurate and robust analysis and interpretation of data[11].

### 4.1.1 How Uniform Sampling is done?

A process known as uniform random sampling is used to arbitrarily and impartially select a smaller group of objects or people from a larger population. The following are the steps involved in conducting a uniform random sample:

- Define the population: Create a clear definition of the population from which you intend to select a representative sample.
- Determine sample size: Determine the ideal sample size that you wish to draw from the population.
- Assign identifiers: Each element in the population should have a unique identification, hence these identifiers must be sequential and begin at a specific number.
- Generate Random Numbers: Use a random number generator or random selection method and the produced random numbers ought to be equally likely to be chosen and evenly distributed.
- Select Sample: Select from the population the objects or people that correspond to the chosen random numbers. Continue doing this until you get the desired number of samples.
- Conduct Analysis: Analyze the selected sample using the appropriate statistical techniques, or run experiments, surveys, etc.



## 4.2 Row Norm Sampling

Row Norm Sampling is a technique used in the field of machine learning and data analysis for selecting a subset of data points from a larger dataset based on their row-wise norms. It aims to prioritize and sample data points that have larger norms or magnitudes, which can be indicative of their importance or relevance in the dataset.

By computing the Euclidean norm or another norm measure of the values in each data point's associated row, Row Norm Sampling determines the row norm for each data point. The row norm denotes the size or length of the vector created by the values in the row.

The data points are arranged in decreasing order according to their row norms as part of the sampling procedure. In comparison to data points with smaller row norms, those with bigger row norms are therefore chosen with a higher likelihood. Data points with greater magnitudes or norms will have a better probability of being included in the sample thanks to this probabilistic selection procedure.

Numerous situations call for the usage of row norm sampling. For instance, Row Norm Sampling can assist in the selection of possible anomalies for further analysis in anomaly identification activities, where outliers or abnormal data points frequently have larger magnitudes relative to normal data points. It can also be helpful when focusing on highly influential observations or when bigger row norms are of particular interest or importance, such as when choosing features.

This method offers a mechanism to collect and prioritize data items based on their relative magnitudes within the dataset by introducing row norms into the sampling process. This enables effective sampling techniques that may be customized to meet particular needs or goals and reveal significant or significant data points.

### 4.2.1 How Row Norm Sampling is Done?

Row Norm Sampling is a technique used to select a subset of data points from a dataset based on their row-wise norms. The process of conducting Row Norm Sampling involves the following steps:

- **Define Dataset:** The dataset from which you want to do Row Norm Sampling should be specified in detail. Multiple data points are grouped in rows and columns in the dataset.
- **Compute Row Norms:** For each data point in the dataset, compute the row norms. Different norm measures, such as the Euclidean norm or the L1 norm, can be used to determine the row norm. It shows the size or length of the vector created by the corresponding row's values.
- **Sort Data points:** The data points are arranged in descending order according to their row norms. From the greatest row norms to the lowest row norms, the data points are arranged in this step.
- **Determine Sampling Probability:** Based on its row norm, assign a sampling probability to each data point. It is possible to proportionally assign the probability so that data points with higher row norms are more likely to be included in the sample.
- **Select Datapoints:** Use a random sampling technique to choose data points from the dataset in accordance with the prescribed sampling probabilities, such as simple random sampling or stratified random sampling. The chosen sample size should be in line with the amount of data points.
- **Perform Analysis:** To get insights or create judgments based on the sampled subset, analyze the selected data points using appropriate statistical techniques, machine learning algorithms, or any other pertinent approaches.

### 4.3 Leverage Score Sampling

The calculation of leverage scores involves the use of matrix factorization techniques, particularly the singular value decomposition (SVD) of the data matrix[12]. The leverage score of a data point is derived from the singular vectors obtained during the SVD process. It is nothing but the, Squared norms of rows of the orthogonal basis of the data matrix[12].

The sampling process involves sorting the data points based on their leverage scores in descending order. Data points with higher leverage scores are given a higher probability of being selected for the sample, while those with lower scores have a lower probability[13]. This probabilistic selection ensures that data points with greater influence on the dataset's structure have a higher chance of being included in the sample.

For example, the leverage score of an observation  $i$  is calculated as the  $i$ -th diagonal element of the hat matrix ( $H$ ), which is derived from the design matrix ( $X$ ) used in the regression model[14].

The leverage score ( $h_i$ ) for the  $i$ -th observation can be calculated as:

$$h_i = X_i(X^T X)^{-1} X_i^T$$

where  $X_i$  is the  $i$ -th row of the design matrix  $X$ [14]. The hat matrix ( $H$ ) is then constructed using the leverage scores, with  $H = X(X^T X)^{-1} X^T$ . Leverage scores provide a measure of the influence that each data point has on the overall structure of the dataset. These scores capture the importance of each data point in terms of its contribution to the variance or structure of the dataset[15].

Leverage Score can be calculated generally by following steps, Let's say we have a data matrix  $X$ , where each row represents a data point and each column represents a feature or attribute.

1. Perform the singular value decomposition (SVD) on the data matrix  $X$ . The SVD decomposes the data matrix into three matrices:  $U$ ,  $\Sigma$ , and  $V^T$ .  
 $U$ : The left singular vectors matrix. Each column of  $U$  represents an orthogonal basis vector that spans the column space of  $X$ .  
 $\Sigma$ : The diagonal matrix of singular values. It contains the singular values associated with each basis vector in  $U$ .  
 $V^T$ : The transpose of the right singular vectors matrix. Each row of  $V^T$  represents an orthogonal basis vector that spans the row space of  $X$ [15].

2. Then the leverage score of each data point is calculated as the squared Euclidean norm of the corresponding row in the matrix  $U$ [15]. In other words, for each row  $i$ , the leverage score  $L_i$  is computed as:

$$L_i = |U_i|^2$$

where  $|U_i|^2$  represents the squared norm of the  $i$ -th row of  $U$ .

Leverage score sampling is an intriguing method for performing approximate computations for big matrices. In fact, it enables the creation of accurate approximations with a level of complexity appropriate for the given issue. However, carrying out leverage scores sampling is a difficult task in and of itself that necessitates further estimates[16].

Leverage Score Sampling is particularly useful in scenarios where the dataset contains outliers, influential observations, or data points that significantly impact the analysis or modeling results. By focusing on data points with higher leverage scores, this technique allows for the selection of influential observations that play a crucial role in capturing the dataset's underlying patterns and characteristics[16].

By incorporating leverage scores into the sampling process, Leverage Score Sampling provides a way to prioritize data points based on their influence on the dataset's structure. This enables the construction of efficient sampling strategies that can effectively capture important and influential observations within the dataset, leading to more accurate analysis and modeling results[16]. Despite these advantageous characteristics, applying leverage score sampling presents a problem in and of itself because it is as complex as an eigendecomposition of the original matrix.

It is important to note that leverage score sampling requires careful consideration of the specific analysis or modeling task, as well as the underlying assumptions and properties of the data. Proper validation and evaluation procedures should be employed to ensure the reliability and validity of the selected subset of data points[17].

## 4.4 K-Medoid Sampling

K-Medoid Sampling is a technique used in data analysis and clustering to select a representative subset of data points from a larger dataset. It is an extension of the K-Means clustering algorithm that focuses on selecting actual data points, known as medoids, as representatives of the clusters.

The process of K-Medoid Sampling involves the following steps:

- **Define the Dataset:** The dataset that you want to analyze and execute K-Medoid Sampling on must be specified precisely. The dataset should have numerous data items with related qualities or features.
- **Define the No. of Clusters:** Enter the appropriate number of clusters (indicated by the letter "K") that you want to locate in the dataset. The number of clusters should be chosen based on the goals of your investigation or the needs of a particular domain.
- **Initialize Medoids:** Choose K data points at random to serve as the starting medoids in the dataset. These medoids will act as the clusters' initial representatives.
- **Assign Data Points to Medoids:** Use a suitable distance metric, such as Euclidean distance, to determine the distance or dissimilarity between each data point and the medoids. Establish the first clusters by associating each data point with the closest medoid.
- **Update Medoids:** Analyse the overall dissimilarity or distance between each data point and all other points in the same cluster for each cluster. Select the new medoid for that cluster based on the data point with the lowest overall dissimilarity.
- **Repeat Steps 4 and 5:** Steps 4 and 5 should be repeated until convergence is obtained, which happens when the medoids stop changing or after a certain number of rounds.
- **Select Medoids as Sample:** Choose the medoids as the representative subset for your investigation once convergence has been reached. These medoids act as representative data points and convey the central tendencies of the clusters.

Using K-Medoid Sampling, it is possible to choose representative medoids as data points to represent the traits of the dataset's clusters. It is frequently utilized in tasks involving data summary, pattern detection, and grouping analysis. K-Medoid Sampling offers a trustworthy method for gathering crucial information from the dataset and aiding meaningful analysis by focusing on actual data points as medoids.

## CHAPTER 5

# Implementation and Dataset Details

## 5.1 Dataset Details

We utilised Set 1, which consists of 6, 983 test questions and 19, 944 training queries from the Yahoo! Learning to Rank challenge (Chapelle and Chang, 2011).

Each query contains a candidate collection of documents, which varies in size, that must be rated. The training set and test set each include a total of 473,134 and 165,660 documents, respectively.

A 700-dimensional feature vector serves as the representative for the query-document pairings. Each query-document combination is given an integer relevance judgment ranging from 0 (poor) to 4(perfect) for supervision.

## 5.2 Implementations Details

We basically do the Empirical Study Sampling techniques for the Fairness Algorithm i.e. "Fair-PG-Rank" given in the paper[2] and check for the output and timestamp for training the sampled dataset and full training dataset.

So initially we do training for the Four sampling techniques that are Uniform Sampling, Row Norm Sampling, Leverage score, and K-medoid sampling taking the sample size from the training dataset of Yahoo LTR dataset(300,500,1000,2500,5000) for both sampling Techniques. Then we train for these sample sizes for all four sampling techniques and analyze the output and timestamp for training the samples for these sampling techniques.

- Firstly, We train for the Set-1 training dataset of the Yahoo Learn to Rank. In we train for the full training dataset of a linear model and all the weights

were randomly initialized between  $(-0.001, 0.001)$  for the linear model.

- an Adam optimizer with a learning rate of 0.001, we set the entropy regularization constant to  $\gamma = 1.0$ , use a baseline, and use a sample size of  $S = 10$  to estimate the gradient and the model is trained for 20 epochs over the training dataset, updating the model one query at a time. And then we check for the output and Time stamp of the model to train the full dataset.
- Now we train for the Row Norm sampling technique for the 300 sample size and check for the output and timestamp. Similarly, we do this for sampling for all the rest of the sample sizes i.e. (500,1000,2500,5000) that we generated and check for the same.
- Now we train for the Uniform Random Sampling technique for the 300 sample size and check for the output and the time is taken for the training of the 300 sampled data on the linear model. Similarly, we do this for sampling for all the rest of the sample sizes i.e. (500,1000,2500,5000) that we generated and check for the same.
- After this, train the model for the Leverage Score sampling technique for the 300 sample size and check for the output and timestamp. Similarly, we do this for sampling for all the rest of the sample sizes i.e. (500,1000,2500,5000) that we generated and check for the results.
- At last we train for the K-Medoid sampling technique for the 300 sample size and check for the output and timestamp. Similarly, we do this for sampling for all the rest of the sample sizes i.e. (500,1000,2500,5000) that we generated and check for the same.

Also, we are checking for the Timestamps or time taken for training the variable-size samples of different sampling strategies and comparing them with each other and also with the time taken to train the full-size dataset. And then check for the results if we are getting desirable and relevant results using different sampling techniques for different sample sizes as compared to the full data size trained results.



## CHAPTER 6

# Results

In the result section we calculate four standard ranking performance measures that are NDCG, DCG, Average Rank, and ERR respectively for the full training data and then for different sampling techniques on different sample sizes respectively, and check for results and analysis. Firstly we'll describe these evaluation metrics. These performance metrics are used to measure the effectiveness of ranking algorithms.

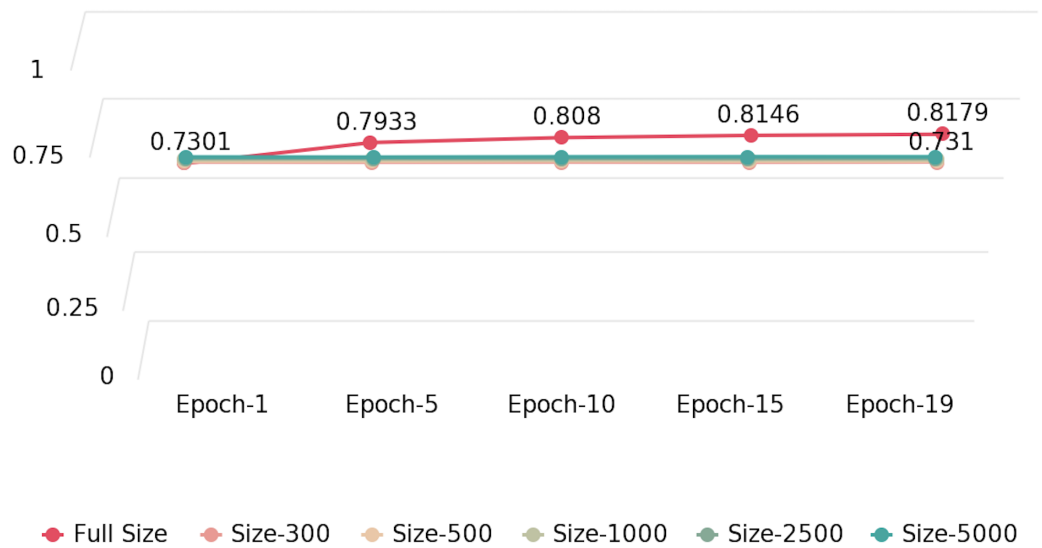
1. DCG: DCG stands for Discounted Cumulative Gain and is a measure of the relevance and ranking of a set of items or documents.
2. NDCG: NDCG stands for Normalized Discounted Cumulative Gain and is a normalized version of DCG. It is calculated by dividing the DCG by the maximum possible DCG for the same set of items.
3. Average Rank: Average Rank measures the average position of relevant items in the ranked list.
4. ERR: ERR stands for Expected Reciprocal Rank. It measures the probability that the user will stop looking at the list after encountering a relevant item.

## 6.1 Uniform Sampling Results

### 6.1.1 NDCG Score

NORMALISED DISCOUNTED CUMULATIVE GAIN(NDCG)						
EPOCH	FULL DATA SIZE-19,944	UNIFORM SAMPLING				
		SIZE-300	SIZE-500	SIZE-1000	SIZE-2500	SIZE-5000
0	0.7301	0.7307	0.73016	0.7306	0.7309	0.7315
5	0.7933	0.7303	0.7301	0.7312	0.7307	0.7311
10	0.8080	0.7307	0.7312	0.7305	0.7309	0.7319
15	0.8146	0.7303	0.7304	0.7308	0.7305	0.7322
19	0.8179	0.7310	0.7303	0.7304	0.7309	0.7320

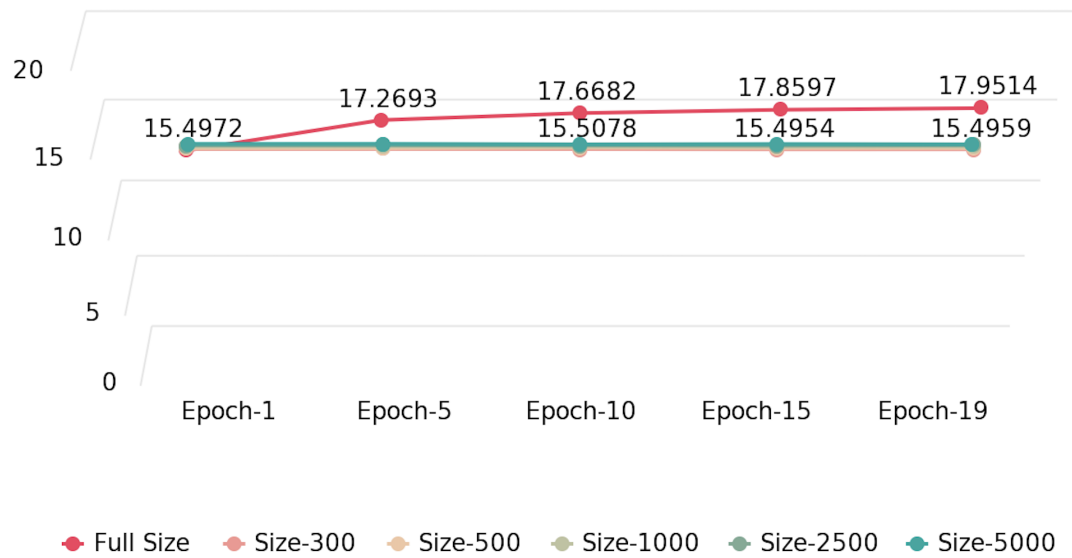
### NDCG Scores For Uniform Sampling



## 6.1.2 DCG Score

DISCOUNTED CUMULATIVE GAIN(DCG)						
EPOCH	FULL DATA SIZE-19,944	UNIFORM SAMPLING				
		SIZE-300	SIZE-500	SIZE-1000	SIZE-2500	SIZE-5000
0	15.4972	15.5092	15.4972	15.5122	15.5105	15.5220
5	17.2693	15.5115	15.4954	15.5161	15.5140	15.5235
10	17.6682	15.5078	15.5214	15.5094	15.5062	15.4976
15	17.8597	15.4954	15.5012	15.5051	15.4984	15.5141
19	17.9514	15.5007	15.4959	15.5130	15.5160	15.4983

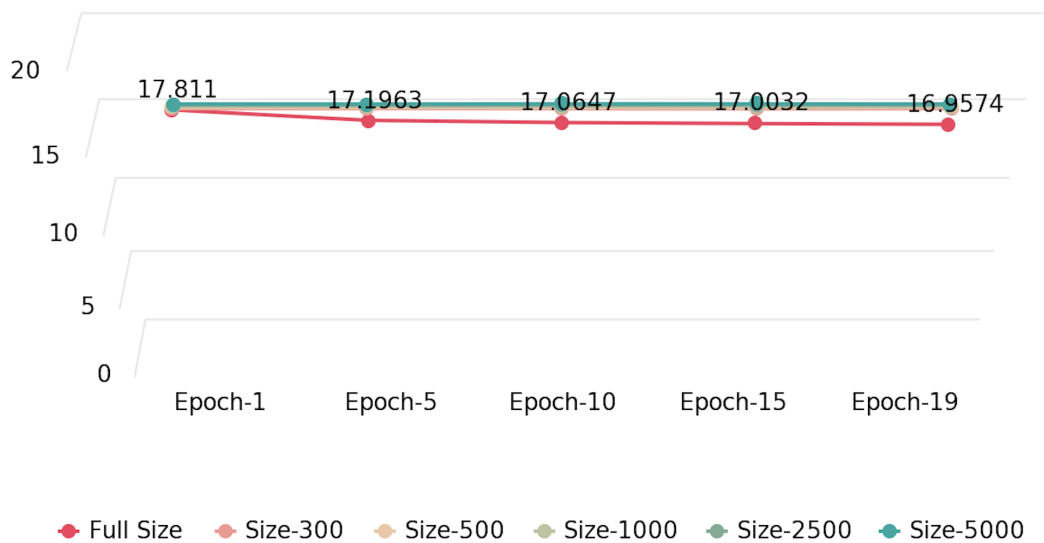
DCG Scores For Uniform Sampling



### 6.1.3 Average Ranking Measure

AVERAGE RANKING MEASURE						
EPOCH	FULL DATA SIZE-19,944	UNIFORM SAMPLING				
		SIZE-300	SIZE-500	SIZE-1000	SIZE-2500	SIZE-5000
0	17.8110	17.8109	17.8203	17.8122	17.8118	17.8082
5	17.1963	17.8155	17.8138	17.8134	17.8093	17.8013
10	17.0647	17.8118	17.8093	17.8183	17.8060	17.8174
15	17.0032	17.8048	17.8130	17.8069	17.8167	17.8151
19	16.9574	17.8131	17.8237	17.8005	17.8123	17.8037

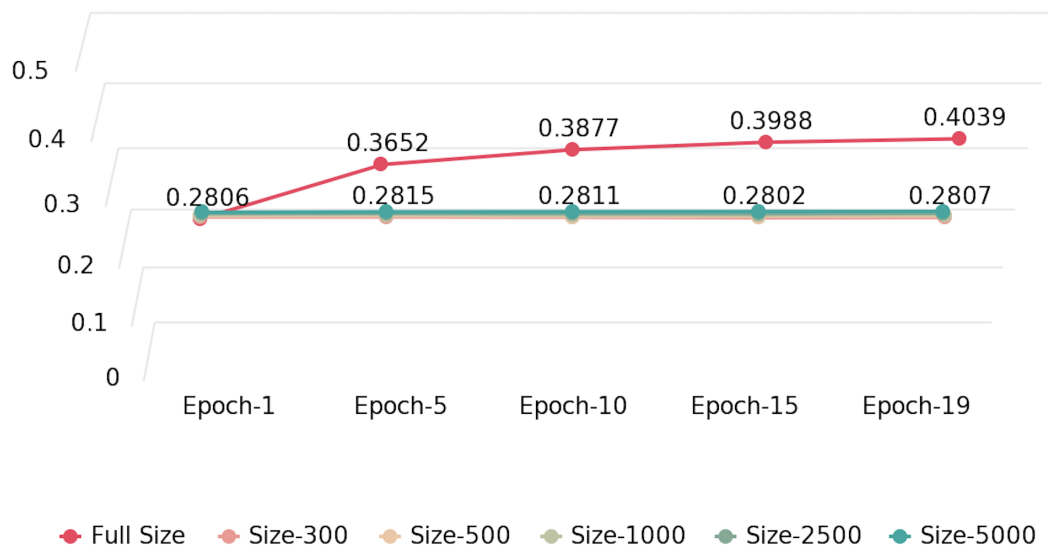
Average Ranking Measure For Uniform Sampling



## 6.1.4 ERR Score

EXPECTED RECIPROCAL RANK(ERR)						
EPOCH	FULL DATA SIZE-19,944	UNIFORM SAMPLING				
		SIZE-300	SIZE-500	SIZE-1000	SIZE-2500	SIZE-5000
0	0.2806	0.2814	0.2806	0.2813	0.2813	0.2812
5	0.3652	0.2815	0.2804	0.2817	0.2816	0.2819
10	0.3877	0.2811	0.2820	0.2810	0.2811	0.2821
15	0.3988	0.2802	0.2807	0.2809	0.2809	0.2824
19	0.4039	0.2807	0.2805	0.2815	0.2817	0.2826

### ERR For Uniform Sampling



## 6.2 Row Norm Sampling Results

### 6.2.1 NDCG Score

NORMALISED DISCOUNTED CUMULATIVE GAIN(NDCG)						
EPOCH	FULL DATA SIZE-19,944	ROW NORM SAMPLING				
		SIZE-300	SIZE-500	SIZE-1000	SIZE-2500	SIZE-5000
0	0.7301	0.7304	0.7309	0.7306	0.7306	0.7313
5	0.7933	0.7308	0.7306	0.7308	0.7306	0.7314
10	0.8080	0.7306	0.7306	0.7308	0.7302	0.7308
15	0.8146	0.7308	0.7304	0.7306	0.7309	0.7308
19	0.8179	0.7301	0.7305	0.7299	0.7307	0.7302

Figure 6.1: NDCG scores for Row Norm Samplings

### 6.2.2 DCG Score

DISCOUNTED CUMULATIVE GAIN(DCG)						
EPOCH	FULL DATA SIZE-19,944	ROW NORM SAMPLING				
		SIZE-300	SIZE-500	SIZE-1000	SIZE-2500	SIZE-5000
0	15.4972	15.5098	15.5149	15.5139	15.4928	15.5224
5	17.2693	15.5122	15.5055	15.514	15.5164	15.5231
10	17.6682	15.5074	15.4996	15.5050	15.5102	15.4978
15	17.8597	15.5098	15.5101	15.5017	15.5076	15.5143
19	17.9514	15.4981	15.5025	15.5010	15.5107	15.4985

Figure 6.2: DCG scores for Row Norm Samplings

### 6.2.3 Average Ranking Measure

AVERAGE RANK MEASURE						
EPOCH	FULL DATA SIZE-19,944	ROW NORM SAMPLING				
		SIZE-300	SIZE-500	SIZE-1000	SIZE-2500	SIZE-5000
0	17.8110	17.8076	17.8109	17.8109	17.8147	17.8086
5	17.1963	17.8186	17.8080	17.8080	17.7977	17.8014
10	17.0647	17.8145	17.8117	17.8117	17.8116	17.8171
15	17.0032	17.8130	17.8175	17.8175	17.8100	17.8159
19	16.9574	17.8095	17.8164	17.8164	17.8029	17.8036

Figure 6.3: Average Rank Measures for Row Norm Samplings

### 6.2.4 ERR Score

EXPECTED RECIPROCAL RANK(ERR)						
EPOCH	FULL DATA SIZE-19,944	ROW NORM SAMPLING				
		SIZE-300	SIZE-500	SIZE-1000	SIZE-2500	SIZE-5000
0	0.2806	0.2812	0.2816	0.2815	0.2804	0.2822
5	0.3652	0.2817	0.2812	0.2814	0.2818	0.2820
10	0.3877	0.2814	0.2807	0.2810	0.2813	0.2809
15	0.3988	0.2814	0.2811	0.2808	0.2811	0.2813
19	0.4039	0.2805	0.2807	0.2808	0.2812	0.2811

Figure 6.4: ERR scores for Row Norm Samplings



## 6.3 Leverage Score Sampling Results

### 6.3.1 NDCG Score

NORMALISED DISCOUNTED CUMULATIVE GAIN(NDCG)						
EPOCH	FULL DATA SIZE-19,944	LEVERAGE SCORE SAMPLING				
		SIZE-300	SIZE-500	SIZE-1000	SIZE-2500	SIZE-5000
0	0.7301	0.7301	0.7307	0.7314	0.7307	0.7306
5	0.7933	0.7302	0.7313	0.7305	0.7303	0.7302
10	0.8080	0.7307	0.7308	0.7308	0.7304	0.7306
15	0.8146	0.7306	0.7300	0.7304	0.7304	0.7303
19	0.8179	0.7308	0.7306	0.7309	0.7307	0.7310

Figure 6.5: NDCG scores for Leverage Score Samplings

### 6.3.2 DCG Score

DISCOUNTED CUMULATIVE GAIN(DCG)						
EPOCH	FULL DATA SIZE-19,944	LEVERAGE SCORE SAMPLING				
		SIZE-300	SIZE-500	SIZE-1000	SIZE-2500	SIZE-5000
0	15.4972	15.5007	15.5090	15.5177	15.5068	15.5133
5	17.2693	15.5025	15.5227	15.5026	15.4995	15.5067
10	17.6682	15.5127	15.5044	15.5158	15.5024	15.5053
15	17.8597	15.5101	15.5041	15.4971	15.5102	15.4927
19	17.9514	15.5085	15.5079	15.5083	15.5100	15.5099

Figure 6.6: DCG scores for Leverage Score Samplings



### 6.3.3 Average Ranking Measure

AVERAGE RANKING MEASURE						
EPOCH	FULL DATA SIZE-19,944	LEVERAGE SCORE SAMPLING				
		SIZE-300	SIZE-500	SIZE-1000	SIZE-2500	SIZE-5000
0	17.8110	17.8104	17.8110	17.8141	17.8074	17.8087
5	17.1963	17.8057	17.8150	17.8008	17.8159	17.8010
10	17.0647	17.8093	17.8060	17.8190	17.8024	17.8211
15	17.0032	17.8113	17.8094	17.8017	17.8092	17.8158
19	16.9574	17.8161	17.8159	17.8084	17.8133	17.8081

Figure 6.7: Average Ranking Measures for Leverage Score Samplings

### 6.3.4 ERR Score

EXPECTED RECIPROCAL RANK(ERR)						
EPOCH	FULL DATA SIZE-19,944	LEVERAGE SCORE SAMPLING				
		SIZE-300	SIZE-500	SIZE-1000	SIZE-2500	SIZE-5000
0	0.2806	0.2807	0.2812	0.2820	0.2814	0.2815
5	0.3652	0.2808	0.2819	0.2812	0.2809	0.2811
10	0.3877	0.2814	0.2810	0.2816	0.2809	0.2811
15	0.3988	0.2813	0.2809	0.2806	0.2808	0.2806
19	0.4039	0.2811	0.2811	0.2813	0.2814	0.2813

Figure 6.8: ERR scores for Leverage Score Samplings

## 6.4 K-Medoid Sampling Results

### 6.4.1 NDCG Score

NORMALISED DISCOUNTED CUMULATIVE GAIN(NDCG)						
EPOCH	FULL DATA SIZE-19,944	K-MEDOID SAMPLING				
		SIZE-300	SIZE-500	SIZE-1000	SIZE-2500	SIZE-5000
0	0.7301	0.7307	0.7308	0.7303	0.7303	0.7303
5	0.7933	0.7312	0.7313	0.7306	0.7307	0.7304
10	0.8080	0.7307	0.7309	0.7309	0.7302	0.7308
15	0.8146	0.7304	0.7305	0.7296	0.7306	0.7304
19	0.8179	0.7309	0.7302	0.7302	0.7307	0.7311

Figure 6.9: NDCG scores for K-medoid Samplings

### 6.4.2 DCG Score

DISCOUNTED CUMULATIVE GAIN(DCG)						
EPOCH	FULL DATA SIZE-19,944	K-MEDOID SAMPLING				
		SIZE-300	SIZE-500	SIZE-1000	SIZE-2500	SIZE-5000
0	15.4972	15.5095	15.5047	15.5030	15.5087	15.5121
5	17.2693	15.5118	15.5165	15.5090	15.5125	15.5164
10	17.6682	15.5084	15.5119	15.5230	15.4959	15.5167
15	17.8597	15.5160	15.5081	15.5018	15.5115	15.6112
19	17.9514	15.5149	15.5084	15.5049	15.5037	15.6126

Figure 6.10: DCG scores for K-medoid Samplings

### 6.4.3 Average Ranking Measure

AVERAGE RANKING MEASURE						
EPOCH	FULL DATA SIZE-19,944	K-MEDOID SAMPLING				
		SIZE-300	SIZE-500	SIZE-1000	SIZE-2500	SIZE-5000
0	17.8110	17.8053	17.8037	17.8161	17.8055	17.8086
5	17.1963	17.8071	17.8060	17.8089	17.8091	17.8010
10	17.0647	17.8120	17.8095	17.8079	17.8136	17.8005
15	17.0032	17.8162	17.8066	17.8122	17.8121	17.7869
19	16.9574	17.8136	17.8036	17.8077	17.8120	17.7688

Figure 6.11: Average Ranking Measures for K-medoid Samplings

### 6.4.4 ERR Score

EXPECTED RECIPROCAL RANK(ERR)						
EPOCH	FULL DATA SIZE-19,944	K-MEDOID SAMPLING				
		SIZE-300	SIZE-500	SIZE-1000	SIZE-2500	SIZE-5000
0	0.2806	0.2813	0.2812	0.2809	0.2812	0.2816
5	0.3652	0.2813	0.2819	0.2811	0.2815	0.2818
10	0.3877	0.2814	0.2815	0.2819	0.2805	0.2822
15	0.3988	0.2817	0.2813	0.2808	0.2812	0.2821
19	0.4039	0.2816	0.2814	0.2808	0.2811	0.2826

Figure 6.12: ERR scores for K-medoid Samplings

## 6.5 Time Stamp Result

Time Stamp Table For Various Experiments					
Sample Sizes	Full Training Data Size-19,944	Time Taken To Run 20 Epochs			
		Uniform Random sampling	Row Norm Sampling	Leverage Score sampling	K-Medoid Sampling
300	Time Taken To Run the Full Training Dataset For 20 Epochs =  <b>13 hrs 42 min 36 sec</b>	41 min 33 sec	43 min 46 sec	41 min 31 sec	40 min 23 sec
500		59 min 31 sec	58 min 44 sec	59 min 50 sec	56 min 14 sec
1000		1 hr 4min 41sec	1 hr 2min 52sec	59min 53sec	59min 52sec
2500		2 hrs 11min 10sec	2 hrs 16 min 10 sec	2hrs 22sec	2hrs 2min 10sec
5000		2 hrs 18 min 21 sec	2 hrs 06 min 27 sec	2 hrs 23 min 32 sec	2 hrs 12 min 36 sec

Figure 6.13: Time Stamps for Various Sample Sizes for Different Sampling Techniques

From the above results, we can clearly observe that we are getting all the standard ranking performance measures which are NDCG, DCG, Average Rank, and ERR somehow nearly equal for the different sampling techniques for different sample sizes as compared to the original full data size trained results.

We can clearly notice in the above Time Stamp table fig(6.17) that for training the full-size dataset the time taken is very large i.e. more than 13 hours but also on the other hand by using different sampling techniques and variable sample sizes of the full dataset the time taken is very small as compared to the time taken by full training dataset i.e. more than 13 hours, also we are getting near equal results as of original full-size dataset trained result.

## CHAPTER 7

# Conclusion and Future work

In conclusion, this empirical study on sampling heuristics for fairness in ranking has shed light on the importance of selecting appropriate sampling methods to ensure fairness in ranking algorithms. We have gained valuable insights into their strengths and limitations by evaluating and comparing different sampling heuristics.

We have found that sampling methods such as uniform random sampling, row norm sampling, leverage score sampling, and K-Medoid Sampling all perform reasonably well in terms of various evaluation measures for the fair PG Rank algorithm. They all give significant time benefits also. It is crucial to keep investigating novel sampling strategies, assessing their efficiency, and comprehending how they affect the impartiality of ranking algorithms. Deriving theoretical bounds and guarantees for various sampling algorithms for Fair PG Rank and other fairness-based algorithms is an important open question.

## References

- [1] Fairness In Ranking ranking description. <https://recsysml.substack.com/p/fairness-in-ranking>. Accessed: 2010-09-30.
- [2] Ashudeep Singh and Thorsten Joachims. Policy learning for fairness in ranking. *Advances in neural information processing systems*, 32, 2019.
- [3] Evaggelia Pitoura, Kostas Stefanidis, and Georgia Koutrika. Fairness in rankings and recommendations: an overview. *The VLDB Journal*, pages 1–28, 2022.
- [4] Ninareh Mehrabi, Fred Morstatter, Nripsuta Saxena, Kristina Lerman, and Aram Galstyan. A survey on bias and fairness in machine learning. *ACM Computing Surveys (CSUR)*, 54(6):1–35, 2021.
- [5] Yunqi Li, Hanxiong Chen, Shuyuan Xu, Yingqiang Ge, Juntao Tan, Shuchang Liu, and Yongfeng Zhang. Fairness in recommendation: A survey. *arXiv preprint arXiv:2205.13619*, 2022.
- [6] Ashudeep Singh and Thorsten Joachims. Fairness of exposure in rankings. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 2219–2228, 2018.
- [7] Yuta Saito and Thorsten Joachims. Fair ranking as fair division: Impact-based individual fairness in ranking. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, page 1514–1524, 2022.
- [8] Meike Zehlike, Ke Yang, and Julia Stoyanovich. Fairness in ranking: A survey. *arXiv preprint arXiv:2103.14000*, 2021.
- [9] Ashudeep Singh, David Kempe, and Thorsten Joachims. Fairness in ranking under uncertainty. *Advances in Neural Information Processing Systems*, 34:11896–11908, 2021.

- [10] Sarjinder Singh and Sarjinder Singh. Simple random sampling. *Advanced Sampling Theory with Applications: How Michael 'selected' Amy Volume I*, pages 71–136, 2003.
- [11] Anita S Acharya, Anupam Prakash, Pikee Saxena, and Aruna Nigam. Sampling: Why and how of it. *Indian Journal of Medical Specialties*, 4(2):330–333, 2013.
- [12] Ali Eshragh, Fred Roosta, Asef Nazari, and Michael W Mahoney. Lsar: Efficient leverage score sampling algorithm for the analysis of big time series data. 2022.
- [13] Brett W Larsen and Tamara G Kolda. Practical leverage-based sampling for low-rank tensor decomposition. *SIAM Journal on Matrix Analysis and Applications*, 43(3):1488–1517, 2022.
- [14] Dimitris Papailiopoulos, Anastasios Kyriillidis, and Christos Boutsidis. Provable deterministic leverage score sampling. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 997–1006, 2014.
- [15] Jason D Lee, Ruoqi Shen, Zhao Song, Mengdi Wang, et al. Generalized leverage score sampling for neural networks. *Advances in Neural Information Processing Systems*, 33:10775–10787, 2020.
- [16] Alessandro Rudi, Daniele Calandriello, Luigi Carratino, and Lorenzo Rosasco. On fast leverage score sampling and optimal learning. *Advances in Neural Information Processing Systems*, 31, 2018.
- [17] Naman Agarwal, Sham Kakade, Rahul Kidambi, Yin-Tat Lee, Praneeth Nectrapalli, and Aaron Sidford. Leverage score sampling for faster accelerated regression and erm. In *Algorithmic Learning Theory*, pages 22–47. PMLR, 2020.