OPT2023: 15th Annual Workshop on Optimization for Machine Learning

Optimizing Group-Fair Plackett-Luce Ranking Models for Relevance and Ex-Post Fairness

Sruthi Gorantla GORANTLAS@IISC.AC.IN
Indian Institute of Science, Bengaluru, India

Eshaan Bhansali* EBHANSALI@WISC.EDU
University of Wisconsin-Madison, Wisconsin, USA

Amit Deshpande† AMITDESH@MICROSOFT.COM
Microsoft Research, Bengaluru, India

Anand Louis† ANANDL@IISC.AC.IN
Indian Institute of Science, Bengaluru, India

Abstract

In learning-to-rank (LTR), optimizing only the relevance (or the expected ranking utility) can cause representational harm to certain categories of items. We propose a novel objective that maximizes expected relevance only over those rankings that satisfy given representation constraints to ensure ex-post fairness. Building upon recent work on an efficient sampler for ex-post group-fair rankings, we propose a group-fair Plackett-Luce model and show that it can be efficiently optimized for our objective in the LTR framework. Experiments on three real-world datasets show that our algorithm guarantees fairness alongside usually having better relevance compared to the LTR baselines. In addition, our algorithm also achieves better relevance than post-processing baselines, which also ensure ex-post fairness. Further, when implicit bias is injected into the training data, our algorithm typically outperforms existing LTR baselines in relevance.

1. Introduction

Stochastic ranking models have gained popularity in LTR [5, 20, 26], primarily due to off-the-shelf gradient-based methods that can be used to optimize these models efficiently. Further, they provide fairness guarantees that deterministic rankings for LTR cannot, e.g., ensuring that multiple items or groups have an equal (or some guaranteed minimum) probability of appearing at the top. We consider the well-known Plackett-Luce (PL) model as our stochastic ranking model. PL model has been used in many fields, such as statistics [14, 22], psychology [17], social choice theory [25], econometrics [2], amongst others. Recent work has increased the popularity, scope, and efficiency of the PL model in LTR [9, 19, 24]. It is also shown to be robust [4] and effective for exploration in online LTR [20, 21]. Recent work has proposed efficient and practical algorithms, namely, PL-Rank and its variants, for optimizing PL ranking models using estimates of the gradient [18]. In addition to optimizing ranking utility, the PL-Rank algorithm also optimizes *fairness of exposure* – an ex-ante fairness metric [24]. There are two types of fairness guarantees one could ask for in a stochastic ranking: *ex-ante* and *ex-post*. Ex-ante fairness asks for satisfying fairness in expectation,

* Work partially done while the author was a research intern at Microsoft Research, Bengaluru, India.
† Equal Contribution

i.e., before the stochastic ranking model realizes a ranking. In contrast, ex-post fairness requires fairness of the actual ranking after the stochastic ranking model generates one. Due to the inherent randomization in the PL model, ex-post fairness guarantees are more challenging to incorporate in the training process such that the resultant model can be optimized efficiently.

Broadly, the fair ranking algorithms can be divided into two groups: *post-processing* and *in-processing*. Post-processing algorithms process the output of a given ranking model to incorporate group-fairness guarantees about sufficient representation of every group (especially, underprivileged demographic groups) in the top positions or top prefixes [1, 6, 12]. As a result, the underlying ranking model may not be optimized in anticipation of the post-processing. In-processing algorithms, on the other hand, incorporate fairness controls to modify the objective in learning-to-rank [18, 23, 24]. As a consequence, previous work on post-processing algorithms in fair ranking can provide ex-post (actual) guarantees on the group-wise representation in the top ranks [6, 12], whereas in-processing algorithms can only provide ex-ante (expected) guarantees on group-wise exposure [23] or amortized individual fairness [3]. The major drawback of the existing LTR algorithms is that none of them optimize relevance while ensuring that every output ranking satisfies group-wise representation guarantees in the top ranks. Our work aims to address this gap.

Consider a job recommendation platform such as *LinkedIn Talent Search*[1], where a stochastic ranking algorithm determines the order in which potential interview candidates from different demographic groups are recommended to recruiters. Let us say there are candidates from two groups – G_1, a majority group with high merit scores, and G_2, a minority group (usually underprivileged) whose merit scores are underestimated due to biases present in the training data used for LTR. These biases may originate from historical imbalances, social prejudices, or systemic inequalities in the data. The stochastic ranking model must output the top-10 candidates every time a recruiter queries for a list of suitable candidates. Consider a particular stochastic ranking that (1) chooses a group G_1 or G_2 with probability 0.5 each, and (2) shows the top 10 candidates from the group chosen in Step 1. This ensures *equal representation* of both the groups ex-ante because there will be 5 candidates in the top 10 from each group, in expectation. However, none of the rankings output by the stochastic ranking satisfies equal representation ex-post. Such rankings may not be aligned with the ethical and diversity hiring policies of the recruiters (or companies).

The main contribution of our work is a novel objective that maximizes expected relevance computed only over those rankings that satisfy given representation constraints for certain sensitive categories or groups of items. We show that a recent post-processing sampler for ex-post group-fair rankings [13] combined with recent ideas to optimize the group-wise PL model [18, 19] can be used to optimize our Group-Fair-PL model efficiently. As a result, we get the best of both worlds: the efficiency of optimization in a fairness-aware in-processing objective and the ex-post fairness guarantees of post-processing methods. Our experiments on three real-world datasets show that our model guarantees ex-post fairness and achieves higher relevance compared to the baselines. When implicit bias [7] is injected into the training data as a stress test or audit for fair ranking algorithms, our algorithm outperforms existing baselines in fairness and relevance.

2. Ex-Post Fairness in Ranking

Preliminaries. Let $\mathcal{I}$ denote the set of items (or documents). Let $\mathsf{S}_k(\mathcal{I})$ be the set of all k sized permutations of the items in $\mathcal{I}$. In the learning-to-rank setup, for any query q, the goal is to output

1. https://business.linkedin.com/talent-solutions

the top-k ranking of relevant items. Let $R_{q,d}$ be an indicator random variable that takes the value of 1 if item d is relevant to q and 0 otherwise. The probability of d being relevant to q is represented as $\rho_d := P(R_{q,d} = 1)$. Let $\sigma \in \mathsf{S}_k(\mathcal{I})$ represent a ranking and let $\sigma(i)$ represent the item in rank i. We use $\sigma(i : i')$ for any $1 \leqslant i < i' \leqslant k$ to represent the set of items in ranks i to i' included in ranking σ, that is, $\sigma(i : i') := \{\sigma(i), \sigma(i+1), \ldots, \sigma(i')\}$. Note that $\sigma(1 : k)$ represents the items in the ranking as a *set*, whereas σ itself is an ordered representation of this set of items. In the following, we drop q from the notation since, in the rest of the paper, we will be working with a fixed query q. We use π to denote a stochastic ranking model (or policy) and Π to denote the set of all stochastic ranking policies. Then the expected relevance metric for $\pi \in \Pi$ is defined as follows,

$$\mathcal{R}(\pi) := \sum_{\sigma \in \mathsf{S}_k(\mathcal{I})} \pi[\sigma] \sum_{i \in [k]} \theta_i \rho_{\sigma(i)}, \tag{1}$$

where $\theta_i \in \mathbb{R}_{\geqslant 0}$ are the position discounts associated with each rank $i \in [k]$ and $\pi[\sigma]$ represents the probability of sampling σ according to π.

Policy Gradients for Placket-Luce. We use π^{PL} to represent the Plackett-Luce (PL) model. This is a popular stochastic ranking model that, given a prediction model m that predicts log scores $m(d)$ for each item d, samples a ranking $\sigma \in \mathsf{S}_k(\mathcal{I})$ with probability, $\pi^{\text{PL}}[\sigma] := \prod_{i=1}^{k} \frac{e^{m(\sigma(i))}}{\sum_{d \in \mathcal{I} \setminus \sigma(1:i-1)} e^{m(d)}}$. Singh and Joachims [24] have proposed using policy gradients to train a PL ranking model to maximize expected relevance. Oosterhuis [19] has developed a computationally efficient way to compute the gradients of the expected relevance metric for the PL model. As a result, PL models can now be trained efficiently to maximize the expected relevance.

Group-Fair Ranking. Suppose the set of items $\mathcal{I}$ can be partitioned into ℓ groups $\mathcal{I}_1, \mathcal{I}_2, \ldots, \mathcal{I}_\ell$ based on the group membership (based on age, race, gender, etc.). For any integer t, let $[t]$ denote $\{1, 2, \ldots, t\}$. We consider the group fairness constraints in the top-k rankings, where, for each group $j \in [\ell]$, we are given a lower bound L_j and an upper bound U_j on the number of items in the top-k ranking from that group. Let $\mathsf{S}_k^{\text{fair}}(\mathcal{I})$ represent all possible group-fair top-k rankings. That is, $\mathsf{S}_k^{\text{fair}}(\mathcal{I}) := \{\sigma \in \mathsf{S}_k(\mathcal{I}) : L_j \leqslant |\sigma(1:k) \cap \mathcal{I}_j| \leqslant U_j, \forall j \in [\ell]\}$. Let $\mathsf{G}_k(\ell) := [\ell]^k$ represent the set of all *group assignment*s of the top-k rankings for ℓ groups. Let $g : \mathcal{I} \to [\ell]$ be the group membership function for the items. We use $g(\sigma)$ to represent the vector of group memberships of the items in the ranking σ. Note that for any σ, $g(\sigma) \in \mathsf{G}_k(\ell)$. We can then define $\mathsf{G}_k^{\text{fair}}(\ell)$ as the set of group assignments that satisfy the group fairness constraints, $\mathsf{G}_k^{\text{fair}}(\ell) := \left\{ g(\sigma) \in [\ell]^k : \sigma \in \mathsf{S}_k^{\text{fair}}(\mathcal{I}) \right\}$. Then for any $\sigma \in \mathsf{S}_k^{\text{fair}}(\mathcal{I})$, $g(\sigma) \in \mathsf{G}_k^{\text{fair}}(\ell)$.

Definition 1 (Ex-Post Fair Policy) *A policy $\pi \in \Pi$ is ex-post fair iff $\forall \sigma \sim \pi,\ g(\sigma) \notin G_k^{fair}(\ell) \implies \pi[\sigma] = 0$.*

Proposed Optimization Objective. We ask for maximizing expected relevance over ex-post group-fair rankings. Then the fair expected relevance can be written as follows,

$$\mathcal{R}^{\text{fair}}(\pi) := \sum_{\sigma \in \mathsf{S}_k^{\text{fair}}(\mathcal{I})} \pi[\sigma] \sum_{i \in [k]} \theta_i \rho_{\sigma(i)}, \ \text{ if } \pi \text{ is ex-post fair, and } 0 \text{ otherwise.} \tag{2}$$

We note that the PL model may not satisfy ex-post fairness in general. In the next section we describe our ex-post fair policy, from which we can sample group-fair rankings efficiently. As a result, we get an efficient algorithm to compute gradients of our proposed model for optimizing $\mathcal{R}^{\text{fair}}$. We can then use the stochastic gradient descent method to train our model. See Appendix A for a discussion about other approaches for ex-post fair rankings and their limitations.

3. Group-Fair Placektt-Luce Model

Let π^{fair} represent the Group-Fair-PL model we propose. In π^{fair}, we have a two-step process to sample ex-post group-fair rankings, (1) Sample a top-k group assignment $\gamma \in \mathsf{G}_k^{\text{fair}}(\ell)$, and (2) Sample a top-$k$ ranking $\sigma \in \mathsf{S}_k^{\text{fair}}(\mathcal{I})$ such that $g(\sigma) = \gamma$. Then,

$$\pi^{\text{fair}}[\sigma] = \mu[g(\sigma)]\pi^{\text{fair}}[\sigma \mid g(\sigma)], \tag{3}$$

where $\mu[\cdot]$ is the distribution over $\{\gamma \in \mathsf{G}_k^{\text{fair}}(\ell)\}$, and $\pi^{\text{fair}}[\cdot \mid \gamma]$ is a conditional distribution over $\{\sigma \in \mathsf{S}_k^{\text{fair}}(\mathcal{I}) : g(\sigma) = \gamma\}$. It is clear that, to achieve ex-post fairness, we can only sample group-fair group assignments in Step 1. For Step 2, we use PL model for items within the group for the ranks assigned to that group according to γ. Therefore, in the Group-Fair-PL model, any group-fair ranking $\sigma \in \mathsf{S}_k^{\text{fair}}(\mathcal{I})$ is sampled with probability,

$$\pi^{\text{fair}}[\sigma] := \mu[g(\sigma)] \prod_{i=1}^{k} \frac{e^{m(\sigma(i))}}{\sum_{d \in \underbrace{\mathcal{I}_{g(\sigma(i))}}_{\text{items from group}} \setminus \sigma(1:i-1)} e^{m(d)}}, \tag{4}$$

and any non-group-fair ranking is sampled with probability 0. Therefore, $\mathcal{R}^{\text{fair}}$ defined in (2) is always evaluated in the **if** case for our Group-Fair-PL model. Let σ_j be the (sub-) ranking of items from group j in σ. We use π_j^{PL} to represent the group-wise PL model for group j. Note that for any $j, j' \in [\ell]$, σ_j and $\sigma_{j'}$ are sampled independently from π_j^{PL} and $\pi_{j'}^{\text{PL}}$ respectively. Given a group assignment $\gamma \in \mathsf{G}_k^{\text{fair}}(\ell)$, let $\psi_j(\gamma) \subseteq [k]$ be the subset of the ranks assigned to group j according to γ. Since $\mathcal{I}_1, \ldots, \mathcal{I}_\ell$ form a partition of $\mathcal{I}$, $\psi_1(\gamma), \ldots, \psi_\ell(\gamma)$ form a partition of $[k]$. Therefore, Equation 4 can be written as,

$$\pi^{\text{fair}}[\sigma] := \mu[g(\sigma)] \prod_{j=1}^{\ell} \prod_{i \in \psi_j(g(\sigma))} \frac{e^{m(\sigma(i))}}{\sum_{d \in \mathcal{I}_{g(\sigma(i))} \setminus \sigma(1:i-1)} e^{m(d)}} = \mu[g(\sigma)] \prod_{j=1}^{\ell} \pi_j^{\text{PL}}[\sigma_j]. \tag{5}$$

It is easy to see that π^{fair} is a valid probability distribution over $\mathsf{S}_k^{\text{fair}}(\mathcal{I})$ (see Lemma 4 in Appendix B). It remains to understand what should be the distribution $\mu[\cdot]$. The distribution from [13] to sample the fair group assignment γ is efficiently samplable and the gradients in this model are efficiently computable. In fact, the distribution does not depend on the predicted scores of the items. Hence, the computation of the gradients boils down to computation of the gradients for each of the group-wise PL models, which we can do efficiently owing to the PL-Rank-3 algorithm in [19]. Our following result then shows that using the distribution given by [13] for μ in our Group-Fair-PL model gives us an efficient algorithm to compute gradient of $\mathcal{R}^{\text{fair}}$ with respect to the predicted scores m. See Appendix B for the proof. Note that PL-Rank-3 takes time $O\left(M\left(|\mathcal{I}| + k \log |\mathcal{I}|\right)\right)$ to compute the gradients.

Theorem 2 *Algorithm 1 estimates the gradient of the relevance metric $\mathcal{R}^{fair}$ in the Group-Fair-PL model in time* $O\left(Mk^2\ell + M\left(|\mathcal{I}| + k\ell \log |\mathcal{I}|\right)\right)$.

Algorithm 1 Group-Fair-PL

Input: items: $\mathcal{I}$, relevance scores: ρ, ranking metric weights: θ, prediction model: m, number of samples: M, groups: $\mathcal{I}_1, \mathcal{I}_2, \ldots, \mathcal{I}_\ell$, bounds: L, U, ranking model: π^{fair}

Output: Gradients $\frac{\delta}{\delta m}\mathcal{R}^{\text{fair}}(\pi^{\text{fair}})$.

Sample M group assignments $\gamma^{(1)}, \gamma^{(2)}, \ldots, \gamma^{(M)}$ from [13] with fairness constraints L, U.
for each group $j = 1, 2, \ldots, \ell$ **do**
  for $t = 1, 2, \ldots, M$ **do**
    Sample $\sigma_j^{(t)}$ from π_j^{PL} for ranks $\psi_j(\gamma)$.
  end for
  For all $d \in \mathcal{I}$, estimate the gradient $\frac{\delta}{\delta m(d)}\mathcal{R}_j(\pi_j^{\text{PL}})$ with M samples $\sigma_j^{(1)}, \sigma_j^{(2)}, \ldots, \sigma_j^{(M)}$ using PL-Rank-3 from [19] and call it $\zeta_j(d)$.
end for
return gradient estimated for each $d \in \mathcal{I}$ as $\frac{\delta}{\delta m(d)}\mathcal{R}^{\text{fair}}(\pi^{\text{fair}}) = \sum_{j\in[\ell]} \zeta_j(d)$.

Table 1: Parameters and results of the experiments on various datasets.

Dataset							Experiment					
Name	**#queries**	**max #items per query**	**Relevance Labels**	**Sensitive feature**	**Groups**	**Minority**	M (Alg. 1)	k	δ	**Avg. running time (sec.)** (*Group-Fair-PL*)	**Avg. running time (sec.)** (*PL-Rank-3*)	**Reference**
MovieLens	2290	588	$1, 2, 3, 4, 5$	Genre	Action(33%), Crime(12%), Romance(30%), Musical(9%), Sci-Fi(16%)	Crime	10	10	0.02	4285	118	Figure 1
German Credit	500	25	$0, 1$	Gender	Male(74%), Female(26%)	Female	50	20	0.05	3008	59	Figure 2
HMDA (AK)	75	25	$0, 1$	Gender	Male(71%), Female(29%)	Female	100	25	0.06	1528	50	Figure 3 Appendix C
HMDA (CT)	731	100	$0, 1$	Gender	Male(67%), Female(33%)	Female	10	25	0.06	7850	673	Figure 4 Appendix C

4. Experiments

Details of the datasets with hyperparameter settings are in Table 1 and Appendix C.

Metrics and baselines. We use NDCG as the ranking utility metric, with position discounts $\theta_i = \frac{1}{\log_2(i+1)}$ for all $i \in [k]$ (first row in all the figures). The second row in the figures shows the *fraction of rankings* sampled from the stochastic ranking models, where an item from the minority group is placed at rank i for each rank $i \in [k]$. The minority group is as mentioned in Table 1. The lower and upper bound lines in the figures show $(p \pm \delta)k$, where p is the proportion of the minority group in the dataset and δ is a small number (see Table 1). Apart from PL-Rank-3, we consider **PL-Rank-3 + GDL22** and **PL-Rank-3 + GAK19** as baselines to compare our fair in-processing algorithm Group-Fair-PL with post-processing baselines by [13] and [12], respectively. We also compare results with **PL-Rank-3 (true)** which is the PL model trained with PL-Rank-3 on the unbiased (or true) relevance scores.

Group-Fair-PL gets the best of both fairness and NDCG. In the presence of implicit bias, Group-Fair-PL outperforms PL-Rank-3 in the NDCG computed on the true scores and achieves almost same NDCG as PL-Rank-3 (true). Compared to just post-processing for ex-post fairness

(PL-Rank-3 + GDL22 and PL-Rank-3 + GAK19), our algorithm almost always achieves better NDCG, This suggests that by explicitly enforcing ex-post fairness during training, we are able to overcome implicit bias via eliminating unreliable comparisons of items from different groups – main motivation of [13]. Even when there is no bias, our Group-Fair-PL still outputs ex-post-fair rankings while not compromising on the NDCG. Moreover, PL-Rank-3 and PL-Rank-3 + GAK19 push the protected group items to the bottom of the ranking in the presence of bias (see Figure 3 row 2), even when their *true* representation is uniform across the ranks (see PL-Rank-3 (true)).

5. Conclusion

We propose a novel group-fair Plackett-Luce model for stochastic ranking and show how one can optimize it efficiently for high relevance along with ex-post group-fairness instead of ex-ante fairness known in previous literature on fair learning-to-rank. We experimentally validate the fairness and relevance guarantees of our ranking models on real-world datasets. Extending our results to other stochastic ranking models in random utility theory is an important direction for future work.

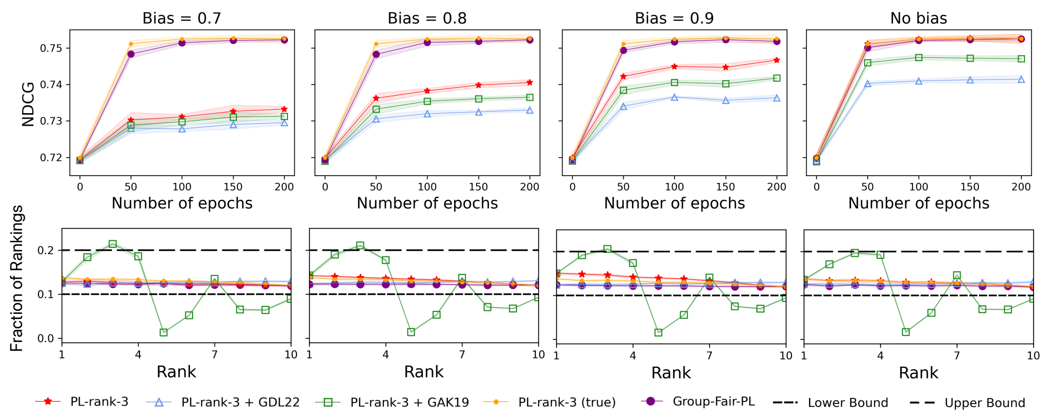

Figure 1: Results on the MovieLens dataset.

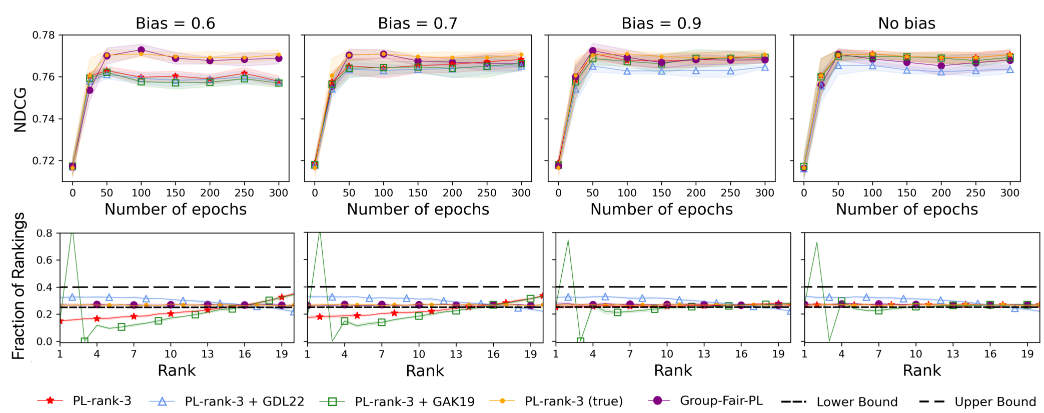

Figure 2: Results on the German Credit dataset.

References

[1] Abolfazl Asudeh, H. V. Jagadish, Julia Stoyanovich, and Gautam Das. Designing fair ranking schemes. In *Proceedings of the 2019 International Conference on Management of Data*, page 1259–1276, 2019.

[2] S Beggs, S Cardell, and J Hausman. Assessing the potential demand for electric cars. *Journal of Econometrics*, 17(1):1–19, 1981.

[3] Asia J. Biega, Krishna P. Gummadi, and Gerhard Weikum. Equity of attention: Amortizing individual fairness in rankings. In *The 41st International ACM SIGIR Conference on Research and Development in Information Retrieval*, page 405–414, 2018.

[4] Sebastian Bruch, Shuguang Han, Michael Bendersky, and Marc Najork. A stochastic treatment of learning to rank scoring functions. In *Proceedings of the 13th International Conference on Web Search and Data Mining*, page 61–69, 2020.

[5] Zhe Cao, Tao Qin, Tie-Yan Liu, Ming-Feng Tsai, and Hang Li. Learning to rank: From pairwise approach to listwise approach. In *Proceedings of the 24th International Conference on Machine Learning*, page 129–136, 2007.

[6] L. Elisa Celis, Damian Straszak, and Nisheeth K. Vishnoi. Ranking with fairness constraints. In *45th International Colloquium on Automata, Languages, and Programming, (ICALP)*, volume 107 of *LIPIcs*, pages 28:1–28:15, 2018.

[7] L. Elisa Celis, Anay Mehrotra, and Nisheeth K. Vishnoi. Interventions for ranking in the presence of implicit bias. In *Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency*, page 369–380, 2020.

[8] A. Feder Cooper, Solon Barocas, Christopher De Sa, and Siddhartha Sen. Variance, Self-Consistency, and Arbitrariness in Fair Classification, 2023.

[9] Fernando Diaz, Bhaskar Mitra, Michael D. Ekstrand, Asia J. Biega, and Ben Carterette. Evaluating stochastic rankings with expected exposure. In *Proceedings of the 29th ACM International Conference on Information and Knowledge Management*, page 275–284, 2020.

[10] Chris Ding and Xiaofeng He. K-means clustering via principal component analysis. In *ICML*, 2004.

[11] Federal Financial Institutions Examination Council. HMDA Data Publication, 2017. URL `https://www.consumerfinance.gov/data-research/hmda/historic-data/`. Released due to the Home Mortgage Disclosure Act.

[12] Sahin Cem Geyik, Stuart Ambler, and Krishnaram Kenthapadi. Fairness-aware ranking in search and recommendation systems with application to linkedin talent search. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, page 2221–2231, 2019.

[13] Sruthi Gorantla, Amit Deshpande, and Anand Louis. Sampling random group fair rankings. *CoRR*, abs/2203.00887, 2022.

[14] Isobel Claire Gormley and Thomas Brendan Murphy. A grade of membership model for rank data. *Bayesian Analysis*, 4(2):265 – 295, 2009.

[15] F. Maxwell Harper and Joseph A. Konstan. The movielens datasets: History and context. *ACM Trans. Interact. Intell. Syst.*, 5(4):19:1–19:19, 2016.

[16] Hans Hofmann. Statlog (German Credit Data). UCI Machine Learning Repository, 1994. DOI: https://doi.org/10.24432/C5NC77.

[17] R. Duncan Luce. *Individual Choice Behavior: A Theoretical analysis*. Wiley, 1959.

[18] Harrie Oosterhuis. Computationally efficient optimization of plackett-luce ranking models for relevance and fairness. In *SIGIR '21: The 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1023–1032, 2021.

[19] Harrie Oosterhuis. Learning-to-rank at the speed of sampling: Plackett-luce gradient estimation with minimal computational complexity. In *SIGIR '22: The 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 2266–2271, 2022.

[20] Harrie Oosterhuis and Maarten de Rijke. Differentiable unbiased online learning to rank. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management (CIKM)*, page 1293–1302, 2018.

[21] Harrie Oosterhuis and Maarten de Rijke. Unifying online and counterfactual learning to rank: A novel counterfactual estimator that effectively utilizes online interventions. In *Proceedings of the 14th ACM International Conference on Web Search and Data Mining*, page 463–471, 2021.

[22] R. L. Plackett. The analysis of permutations. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, 24(2):193–202, 1975.

[23] Ashudeep Singh and Thorsten Joachims. Fairness of exposure in rankings. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD*, pages 2219–2228, 2018.

[24] Ashudeep Singh and Thorsten Joachims. Policy learning for fairness in ranking. In *Advances in Neural Information Processing Systems*, volume 32, 2019.

[25] Hossein Azari Soufiani, David C. Parkes, and Lirong Xia. Random utility theory for social choice. In *Proceedings of the 25th International Conference on Neural Information Processing Systems - Volume 1*, page 126–134, 2012.

[26] Fen Xia, Tie-Yan Liu, Jue Wang, Wensheng Zhang, and Hang Li. Listwise approach to learning to rank: Theory and algorithm. In *Proceedings of the 25th International Conference on Machine Learning*, page 1192–1199, 2008.

[27] Himank Yadav, Zhengxiao Du, and Thorsten Joachims. Policy-gradient training of fair and unbiased ranking functions. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '21, page 1044–1053, New

York, NY, USA, 2021. Association for Computing Machinery. ISBN 9781450380379. doi: 10.1145/3404835.3462953. URL https://doi.org/10.1145/3404835.3462953.

Appendix A. Other Approaches to Train Ex-Post Group-Fair LTR Models

Limitations of Plackett-Luce. There have been two significant contributions toward fair ranking with PL models. We list them and point out their limitations below.

1. **In-processing.** Asudeh et al. [1] and Oosterhuis [18] have proposed policy gradients-based optimization for expected relevance and equity of expected exposure of groups of items for PL models. The major drawback of these methods is that fairness is measured in expectation. Therefore, the trained PL model may not satisfy ex-post fairness.

2. **Post-processing.** Celis et al. [6], Geyik et al. [12], Gorantla et al. [13], Singh and Joachims [23] and many other previous works have proposed algorithms to post-process the scores or the ranking output by any LTR model (or specifically PL) to satisfy fairness. Ex-post fairness is satisfied in this case, but the trained LTR model is unaware of the post-processing that is going to be applied on the scores. Hence, it may end up learning a bad solution.

We overcome these limitations by incorporating ex-post fairness during the training process of the PL-based LTR. Towards this end, we propose to use a different objective function for the stochastic ranking models in Equation 2.

However, we note that we could also optimize a different relevance metric $\widehat{\mathcal{R}}$ defined over $\mathsf{S}_k^{\text{fair}}(\mathcal{I})$,

$$\widehat{\mathcal{R}}(\pi) := \sum_{\sigma \in \mathsf{S}_k^{\text{fair}}(\mathcal{I})} \pi[\sigma] \sum_{i \in [k]} \theta_i \rho_{\sigma(i)}.$$

For any ex-post fair policy π, $\widehat{\mathcal{R}}(\pi) = \mathcal{R}^{\text{fair}}(\pi)$. Moreover, if the fairness constraints are vacuous, that is, $L_j = 0$ and $U_j = k$, for all $j \in [\ell]$, then, $\widehat{\mathcal{R}}(\pi) = \mathcal{R}^{\text{fair}}(\pi) = \mathcal{R}(\pi)$. Note that $\widehat{\mathcal{R}}$ does not strictly enforce ex-post fairness while training. Hence PL model can be trained for optimizing $\widehat{\mathcal{R}}$. One could use *rejection sampling* to enforce ex-post fairness during and after training. That is, to output a ranking from this model, we need to sample rankings from this model until we see a fair ranking. However, in general, the probability of seeing a fair ranking may be very small. For example, if the fairness constraints are such that $L_j = U_j$ for all but a constant number of groups in $[\ell]$, and the predicted scores of the items are such that from each group $j \in [\ell]$, k items have a score of 1 and others have score 0, then the probability of seeing a fair ranking is $\frac{k^c}{k^\ell}$, where c is a constant. This means that, in the PL model, one needs to sample $O(k^\ell)$ many rankings in expectation before seeing a fair ranking[2], which is computationally inefficient. This also affects the training process since the estimate of the gradient only makes sense if we have enough samples that are fair rankings.

For these reasons, asking for stochastic ranking models that can be trained with $\mathcal{R}^{\text{fair}}$ as an objective is well-motivated.

Appendix B. Missing Proofs

The distribution μ from [13] samples a fair group assignment γ in two steps:

1. First, it samples a tuple from the set $\{(x_1, x_2, \ldots, x_\ell) \in [k]^\ell : L_j \leqslant x_j \leqslant U_j \wedge x_1 + x_2 + \cdots + x_\ell = k\}$, uniformly at random. In this tuple, x_j gives the number of items from group j to be sampled in the top-k ranking.

2. This follows from the fact that the expected value of a geometric random variable with parameter $p := \frac{k^c}{k^\ell}$ is $1/p$.

2. Then it samples a group assignment $\gamma = (\gamma_1, \gamma_2, \ldots, \gamma_k)$ to be a uniform random permutation of the vector $(\underbrace{1, 1, \ldots, 1}_{x_1 \text{ times}}, \underbrace{2, 2, \ldots, 2}_{x_2 \text{ times}}, \ldots, \underbrace{\ell, \ell, \ldots, \ell}_{x_\ell \text{ times}})$.

Below, we re-state their theorem about the time taken to sample a fair group assignment from this distribution.

Theorem 3 (Theorem 4.1 in [13]) *There is a dynamic programming-based algorithm that samples a group assignment γ in time $O(k^2\ell)$.*

Therefore, this distribution is efficiently samplable. Moreover, this distribution also satisfies additional desirable properties, which we will discuss further in Section 4.

Lemma 4 *π^{fair} is a valid probability distribution over $\mathsf{S}_k^{fair}(\mathcal{I})$.*

Proof It is clear that $\pi^{\text{fair}}[\sigma] \geqslant 0$ for each $\sigma \in \mathsf{S}_k^{\text{fair}}(\mathcal{I})$. Moreover, non-group-fair rankings are sampled from μ with probability 0. Therefore, $\pi^{\text{fair}}[\sigma] = 0$, for every $\sigma \notin \mathsf{S}_k^{\text{fair}}(\mathcal{I})$. Further,

$$
\begin{aligned}
&\sum_{\sigma \in \mathsf{S}_k^{\text{fair}}(\mathcal{I})} \pi^{\text{fair}}[\sigma] \\
&\qquad = \sum_{\sigma \in \mathsf{S}_k^{\text{fair}}(\mathcal{I})} \mu[g(\sigma)] \prod_{j \in [\ell]} \pi_j^{\text{PL}}[\sigma_j \mid g(\sigma)] \\
&\qquad = \sum_{\gamma \in \mathsf{G}_k^{\text{fair}}(\ell)} \mu[\gamma] \sum_{\substack{\sigma \in \mathsf{S}_k(\mathcal{I}) \\ \text{s.t. } g(\sigma)=\gamma}} \prod_{j \in [\ell]} \pi_j^{\text{PL}}[\sigma_j \mid \gamma] \\
&\qquad = \sum_{\gamma \in \mathsf{G}_k^{\text{fair}}(\ell)} \mu[\gamma] \left(\sum_{\sigma_1 \in \mathsf{S}_{|\psi_1(\gamma)|}(\mathcal{I}_1)} \left(\sum_{\sigma_2 \in \mathsf{S}_{|\psi_2(\gamma)|}(\mathcal{I}_2)} \cdots \left(\sum_{\sigma_\ell \in \mathsf{S}_{|\psi_\ell(\gamma)|}(\mathcal{I}_\ell)} \prod_{j \in [\ell]} \pi_j^{\text{PL}}[\sigma_j \mid \gamma] \right) \right) \right) \\
&\qquad = \sum_{\gamma \in \mathsf{G}_k^{\text{fair}}(\ell)} \mu[\gamma] \left(\sum_{\sigma_1 \in \mathsf{S}_{|\psi_1(\gamma)|}(\mathcal{I}_1)} \pi_1^{\text{PL}}[\sigma_1 \mid \gamma] \left(\sum_{\sigma_2 \in \mathsf{S}_{|\psi_2(\gamma)|}(\mathcal{I}_2)} \pi_2^{\text{PL}}[\sigma_2 \mid \gamma] \cdots \right.\right. \\
&\qquad\qquad\qquad \left.\left. \cdots \left(\sum_{\sigma_{\ell-1} \in \mathsf{S}_{|\psi_\ell(\gamma)|}(\mathcal{I}_\ell)} \pi_\ell^{\text{PL}}[\sigma_\ell \mid \gamma] \right) \right) \right) \\
&\qquad = \sum_{\gamma \in \mathsf{G}_k^{\text{fair}}(\ell)} \mu[\gamma] \left(\sum_{\sigma_1 \in \mathsf{S}_{|\psi_1(\gamma)|}(\mathcal{I}_1)} \pi_1^{\text{PL}}[\sigma_1 \mid \gamma] \right) \left(\sum_{\sigma_2 \in \mathsf{S}_{|\psi_2(\gamma)|}(\mathcal{I}_2)} \pi_2^{\text{PL}}[\sigma_2 \mid \gamma] \right) \cdots \\
&\qquad\qquad\qquad \cdots \left(\sum_{\sigma_{\ell-1} \in \mathsf{S}_{|\psi_\ell(\gamma)|}(\mathcal{I}_\ell)} \pi_\ell^{\text{PL}}[\sigma_\ell \mid \gamma] \right) \\
&\qquad = \sum_{\gamma \in \mathsf{G}_k^{\text{fair}}(\ell)} \mu[\gamma] \prod_{j \in [\ell]} \sum_{\sigma_j \in \mathsf{S}_{\left|\psi_j(\gamma)\right|}(\mathcal{I}_j)} \pi_j^{\text{PL}}[\sigma_j \mid \gamma]
\end{aligned}
$$

$$
\begin{aligned}
&= \sum_{\gamma \in \mathsf{G}_k^{\text{fair}}(\ell)} \mu[\gamma] \prod_{j \in [\ell]} 1 \\
&= 1.
\end{aligned}
$$

■

Proof [Proof of Theorem 2] Note that given a group assignment γ, the probability of sampling an item d at rank i depends only on the items from group $\gamma(i)$ that appear in ranks 1 to $i-1$. In our Group-Fair PL model, only items from group $\gamma(i)$ are sampled in rank i. Let $\psi_j(\gamma)$ represent the set of ranks assigned to group j according to the group assignment γ, for each $j \in [\ell]$. Then,

$$
\begin{aligned}
\mathcal{R}^{\text{fair}}(\pi^{\text{fair}}) &= \sum_{\sigma \in \mathsf{S}_k^{\text{fair}}(\mathcal{I})} \pi^{\text{fair}}[\sigma] \left(\sum_{i \in [k]} \theta_i \rho_{\sigma(i)} \right) \\
&= \sum_{\sigma \in \mathsf{S}_k^{\text{fair}}(\mathcal{I})} \sum_{\gamma \in \mathsf{G}_k^{\text{fair}}(\ell)} \pi^{\text{fair}}[\gamma, \sigma] \left(\sum_{i \in [k]} \theta_i \rho_{\sigma(i)} \right) \\
&= \sum_{\sigma \in \mathsf{S}_k^{\text{fair}}(\mathcal{I})} \sum_{\gamma \in \mathsf{G}_k^{\text{fair}}(\ell)} \mu[\gamma] \cdot \pi^{\text{fair}}[\sigma \mid \gamma] \left(\sum_{i \in [k]} \theta_i \rho_{\sigma(i)} \right) \\
&= \sum_{\gamma \in \mathsf{G}_k^{\text{fair}}(\ell)} \mu[\gamma] \sum_{\sigma \in \mathsf{S}_k^{\text{fair}}(\mathcal{I})} \pi^{\text{fair}}[\sigma \mid \gamma] \left(\sum_{i \in [k]} \theta_i \rho_{\sigma(i)} \right).
\end{aligned}
$$

Equation 5 gives us,

$$
\mathcal{R}^{\text{fair}}(\pi^{\text{fair}}) = \sum_{\gamma \in \mathsf{G}_k^{\text{fair}}(\ell)} \mu[\gamma] \left(\sum_{\sigma \in \mathsf{S}_k^{\text{fair}}(\mathcal{I})} \left(\prod_{j \in [\ell]} \pi_j^{\text{PL}}[\sigma_j \mid \gamma] \right) \left(\sum_{i \in [k]]} \theta_i \rho_{\sigma(i)} \right) \right).
$$

Now since $\psi_1(\gamma), \psi_2(\gamma), \ldots, \psi_\ell(\gamma)$ form a partition of $[k]$ we can rearrange the terms to get the following,

$$
\begin{aligned}
&\mathcal{R}^{\text{fair}}(\pi^{\text{fair}}) \\
&= \sum_{\gamma \in \mathsf{G}_k^{\text{fair}}(\ell)} \mu[\gamma] \left(\sum_{\sigma \in \mathsf{S}_k^{\text{fair}}(\mathcal{I})} \left(\prod_{j \in [\ell]} \pi_j^{\text{PL}}[\sigma_j \mid \gamma] \right) \left(\sum_{j \in [\ell]} \sum_{i \in \psi_j(\gamma)} \theta_i \rho_{\sigma(i)} \right) \right) \\
&= \sum_{\gamma \in \mathsf{G}_k^{\text{fair}}(\ell)} \mu[\gamma] \left(\sum_{\sigma_1 \in \mathsf{S}_{|\psi_1(\gamma)|}(\mathcal{I}_1)} \sum_{\sigma_2 \in \mathsf{S}_{|\psi_2(\gamma)|}(\mathcal{I}_2)} \cdots \right. \\
&\qquad \left. \cdots \sum_{\sigma_\ell \in \mathsf{S}_{|\psi_\ell(\gamma)|}(\mathcal{I}_\ell)} \left(\prod_{j \in [\ell]} \pi_j^{\text{PL}}[\sigma_j \mid \gamma] \right) \left(\sum_{j \in [\ell]} \sum_{i \in \psi_j(\gamma)} \theta_i \rho_{\sigma(i)} \right) \right)
\end{aligned}
$$

$$= \sum_{\gamma \in \mathsf{G}_k^{\text{fair}}(\ell)} \mu[\gamma] \left(\sum_{\sigma_1 \in \mathsf{S}_{|\psi_1(\gamma)|}(\mathcal{I}_1)} \pi_1^{\text{PL}}[\sigma_1 \mid \gamma] \sum_{\sigma_2 \in \mathsf{S}_{|\psi_2(\gamma)|}(\mathcal{I}_2)} \pi_2^{\text{PL}}[\sigma_2 \mid \gamma] \cdots \right.$$
$$\left. \cdots \sum_{\sigma_\ell \in \mathsf{S}_{|\psi_\ell(\gamma)|}(\mathcal{I}_\ell)} \pi_\ell^{\text{PL}}[\sigma_\ell \mid \gamma] \left(\sum_{j \in [\ell]} \sum_{i \in \psi_j(\gamma)} \theta_i \rho_{\sigma(i)} \right) \right).$$

Now the last term can be written as,

$$\sum_{\sigma_\ell \in \mathsf{S}_{|\psi_\ell(\gamma)|}(\mathcal{I}_\ell)} \pi_\ell^{\text{PL}}[\sigma_\ell \mid \gamma] \left(\sum_{j \in [\ell]} \sum_{i \in \psi_j(\gamma)} \theta_i \rho_{\sigma(i)} \right)$$
$$= \sum_{\sigma_\ell \in \mathsf{S}_{|\psi_\ell(\gamma)|}(\mathcal{I}_\ell)} \pi_\ell^{\text{PL}}[\sigma_\ell \mid \gamma] \left(\sum_{i \in \psi_\ell(\gamma)} \theta_i \rho_{\sigma(i)} + \sum_{j \in [\ell-1]} \sum_{i \in \psi_j(\gamma)} \theta_i \rho_{\sigma(i)} \right)$$
$$= \sum_{\sigma_\ell \in \mathsf{S}_{|\psi_\ell(\gamma)|}(\mathcal{I}_\ell)} \pi_\ell^{\text{PL}}[\sigma_\ell \mid \gamma] \left(\sum_{i \in \psi_\ell(\gamma)} \theta_i \rho_{\sigma(i)} \right)$$
$$+ \sum_{j \in [\ell-1]} \sum_{i \in \psi_j(\gamma)} \theta_i \rho_{\sigma(i)} \underbrace{\sum_{\sigma_\ell \in \mathsf{S}_{|\psi_\ell(\gamma)|}(\mathcal{I}_\ell)} \pi_\ell^{\text{PL}}[\sigma_\ell \mid \gamma]}_{=1}$$
$$= \sum_{\sigma_\ell \in \mathsf{S}_{|\psi_\ell(\gamma)|}(\mathcal{I}_\ell)} \pi_\ell^{\text{PL}}[\sigma_\ell \mid \gamma] \left(\sum_{i \in \psi_\ell(\gamma)} \theta_i \rho_{\sigma(i)} \right) + \sum_{j \in [\ell-1]} \sum_{i \in \psi_j(\gamma)} \theta_i \rho_{\sigma(i)}.$$

Taking the summation $\sum_{\sigma_{\ell-1} \in \mathsf{S}_{|\psi_{\ell-1}(\gamma)|}(\mathcal{I}_{\ell-1})} \pi_{\ell-1}^{\text{PL}}[\sigma_{\ell-1} \mid \gamma]$ on both sides we get,

$$\sum_{\sigma_{\ell-1} \in \mathsf{S}_{|\psi_{\ell-1}(\gamma)|}(\mathcal{I}_{\ell-1})} \pi_{\ell-1}^{\text{PL}}[\sigma_{\ell-1} \mid \gamma] \sum_{\sigma_\ell \in \mathsf{S}_{|\psi_\ell(\gamma)|}(\mathcal{I}_\ell)} \pi_\ell^{\text{PL}}[\sigma_\ell \mid \gamma] \left(\sum_{j \in [\ell]} \sum_{i \in \psi_j(\gamma)} \theta_i \rho_{\sigma(i)} \right)$$
$$= \sum_{\sigma_{\ell-1} \in \mathsf{S}_{|\psi_{\ell-1}(\gamma)|}(\mathcal{I}_{\ell-1})} \pi_{\ell-1}^{\text{PL}}[\sigma_{\ell-1} \mid \gamma] \left(\sum_{\sigma_\ell \in \mathsf{S}_{|\psi_\ell(\gamma)|}(\mathcal{I}_\ell)} \pi_\ell^{\text{PL}}[\sigma_\ell \mid \gamma] \left(\sum_{i \in \psi_\ell(\gamma)} \theta_i \rho_{\sigma(i)} \right) \right.$$
$$\left. + \sum_{j \in [\ell-1]} \sum_{i \in \psi_j(\gamma)} \theta_i \rho_{\sigma(i)} \right)$$
$$= \sum_{\sigma_\ell \in \mathsf{S}_{|\psi_\ell(\gamma)|}(\mathcal{I}_\ell)} \pi_\ell^{\text{PL}}[\sigma_\ell \mid \gamma] \left(\sum_{i \in \psi_\ell(\gamma)} \theta_i \rho_{\sigma(i)} \right) \underbrace{\sum_{\sigma_{\ell-1} \in \mathsf{S}_{|\psi_{\ell-1}(\gamma)|}(\mathcal{I}_{\ell-1})} \pi_{\ell-1}^{\text{PL}}[\sigma_{\ell-1} \mid \gamma]}_{=1}$$
$$+ \sum_{\sigma_{\ell-1} \in \mathsf{S}_{|\psi_{\ell-1}(\gamma)|}(\mathcal{I}_{\ell-1})} \pi_{\ell-1}^{\text{PL}}[\sigma_{\ell-1} \mid \gamma] \sum_{j \in [\ell-1]} \sum_{i \in \psi_j(\gamma)} \theta_i \rho_{\sigma(i)}$$

$$= \sum_{\sigma_\ell \in \mathsf{S}_{|\psi_\ell(\gamma)|}(\mathcal{I}_\ell)} \pi_\ell^{\text{PL}}[\sigma_\ell \mid \gamma] \left(\sum_{i \in \psi_\ell(\gamma)} \theta_i \rho_{\sigma(i)} \right) + \sum_{\sigma_{\ell-1} \in \mathsf{S}_{|\psi_{\ell-1}(\gamma)|}(\mathcal{I}_{\ell-1})} \pi_{\ell-1}^{\text{PL}}[\sigma_{\ell-1} \mid \gamma] \left(\sum_{i \in \psi_{\ell-1}(\gamma)} \theta_i \rho_{\sigma(i)} \right) + \sum_{j \in [\ell-2]} \sum_{i \in \psi_j(\gamma)} \theta_i \rho_{\sigma(i)}.$$

Repeating the above until we reach $j \in [1]$ in the last term, we get,

$$\begin{aligned}
&\mathcal{R}^{\text{fair}}(\pi^{\text{fair}}) \\
&= \sum_{\gamma \in \mathsf{G}_k^{\text{fair}}(\ell)} \mu[\gamma] \left(\sum_{\sigma_1 \in \mathsf{S}_{|\psi_1(\gamma)|}(\mathcal{I}_1)} \pi_1^{\text{PL}}[\sigma_1 \mid \gamma] \left(\sum_{i \in \psi_1(\gamma)} \theta_i \rho_{\sigma(i)} \right) + \cdots + \sum_{\sigma_\ell \in \mathsf{S}_{|\psi_\ell(\gamma)|(\mathcal{I}_\ell)}} \pi_\ell^{\text{PL}}[\sigma_\ell \mid \gamma] \left(\sum_{i \in \psi_\ell(\gamma)} \theta_i \rho_{\sigma(i)} \right) \right) \\
&= \sum_{\gamma \in \mathsf{G}_k^{\text{fair}}(\ell)} \mu[\gamma] \left(\sum_{j \in [\ell]} \sum_{\sigma_j \in \mathsf{S}_{|\psi_j(\gamma)|}(\mathcal{I}_j)} \pi_j^{\text{PL}}[\sigma_j \mid \gamma] \left(\sum_{i \in \psi_j(\gamma)} \theta_i \rho_{\sigma(i)} \right) \right) \\
&= \sum_{\gamma \in \mathsf{G}_k^{\text{fair}}(\ell)} \mu[\gamma] \left(\sum_{j \in [\ell]} \mathcal{R}_j(\pi_j^{\text{PL}}) \right), \qquad (6)
\end{aligned}$$

where $R_j(\gamma)$ is the reward obtained from the group-wise Plackett-Luce model π_j^{PL} for the ranks assigned to group j according to the group assignment γ. That is,

$$\mathcal{R}_j(\pi_j^{\text{PL}}) := \sum_{\sigma_j \in \mathsf{S}_{|\psi_j(\gamma)|}(\mathcal{I}_j)} \pi_j^{\text{PL}}[\sigma_j \mid \gamma] \left(\sum_{i \in \psi_j(\gamma)} \theta_i \rho_{\sigma(i)} \right).$$

Now for a fixed an item d, the derivative with respect to the score of d will be,

$$\frac{\delta}{\delta m(d)} \mathcal{R}^{\text{fair}}(\pi^{\text{fair}}) = \frac{\delta}{\delta m(d)} \sum_{\gamma \in \mathsf{G}_k^{\text{fair}}(\ell)} \mu[\gamma] \left(\sum_{j \in [\ell]} \mathcal{R}_j(\pi_j^{\text{PL}}) \right).$$

Since in our group-fair PL model, the group assignment γ is sampled independently of the score $m(d)$, we have

$$\frac{\delta}{\delta m(d)} \mathcal{R}^{\text{fair}}(\pi^{\text{fair}}) = \sum_{\gamma \in \mathsf{G}_k^{\text{fair}}(\ell)} \mu[\gamma] \left(\sum_{j \in [\ell]} \frac{\delta}{\delta m(d)} \mathcal{R}_j(\pi_j^{\text{PL}}) \right) = \mathbb{E}_{\gamma \sim \mu} \left[\sum_{j \in [\ell]} \frac{\delta}{\delta m(d)} \mathcal{R}_j(\pi_j^{\text{PL}}) \right]. \quad (7)$$

Naïvely applying PL-Rank-3. PL-Rank-3 with N samples can estimation the gradient $\frac{\delta}{\delta m(d)}\mathcal{R}_j(\pi_j^{\text{PL}})$, for a fixed γ, in time $O\left(N\left(|\mathcal{I}_j| + k\log|\mathcal{I}_j|\right)\right)$ for group $j \in [\ell]$. Let us say we take M samples to estimate the outer expectation. From Theorem 3 we have that the time taken to sample one group assignment γ is $O\left(k^2\ell\right)$. Therefore, to sample M group assignments, it takes time $O\left(Mk^2\ell\right)$. Then, the total time taken to compute $\frac{\delta}{\delta m(d)}\mathcal{R}^{\text{fair}}(\pi^{\text{fair}})$ will be

$$O\left(M\left(k^2\ell + \sum_{j\in[\ell]} N\left(|\mathcal{I}_j| + k\log|\mathcal{I}_j|\right)\right)\right) = O\left(Mk^2\ell + MN\left(|\mathcal{I}| + k\ell\log|\mathcal{I}|\right)\right). \quad (8)$$

Correctness for $N = 1$. Let $\text{rank}(\sigma, d)$ represent the rank assigned to item d in σ. Then from PL-Rank-3 algorithm in [19] we know that

$$\frac{\delta}{\delta m(d)}\mathcal{R}_j(\pi_j^{\text{PL}}) = \mathbb{E}_{\sigma_j|\gamma}\left[PR^{(j)}_{\sigma,d} + e^{m(d)}\left(\rho_d DR^{(j)}_{\sigma,d} - RI^{(j)}_{\sigma,d}\right)\right], \quad (9)$$

where

$$PR^{(j)}_{\sigma,i} = \sum_{i'=[i+1,k]\cap\psi_j(\gamma)} \theta_{i'}\rho_{\sigma(i')} \quad \text{and} \quad PR^{(j)}_{\sigma,d} = PR^{(j)}_{\sigma,\text{rank}(\sigma,d)},$$

$$RI^{(j)}_{\sigma,i} = \sum_{i'=[i+1,k]\cap\psi_j(\gamma)} \frac{PR^{(j)}_{\sigma,i}}{\sum_{d'\in\mathcal{I}_j\setminus\sigma(1:i-1)} e^{m(d')}} \quad \text{and} \quad RI^{(j)}_{\sigma,d} = RI^{(j)}_{\sigma,\text{rank}(\sigma,d)},$$

$$DR^{(j)}_{\sigma,i} = \sum_{i'=[i+1,k]\cap\psi_j(\gamma)} \frac{\theta_{\sigma,i}}{\sum_{d'\in\mathcal{I}_j\setminus\sigma(1:i-1)} e^{m(d')}} \quad \text{and} \quad DR^{(j)}_{\sigma,d} = DR^{(j)}_{\sigma,\text{rank}(\sigma,d)}.$$

Note that for a fixed ranking σ, PL-Rank-3 computes the term inside the expectation efficiently in time $O(|\mathcal{I}_j| + k\log|\mathcal{I}_j|)$. Hence, even if the position discount values vary between different samples, or if the length of the ranking $|\psi_j(\gamma)|$ changes between different samples, we can still use PL-Rank-3 algorithm to compute the term inside the expectation for each sample independently and efficiently. Therefore, substituting Equation 9 in Equation 7 we get,

$$\begin{aligned}\frac{\delta}{\delta m(d)}\mathcal{R}^{\text{fair}}(\pi^{\text{fair}}) &= \mathbb{E}_{\gamma\sim\mu}\left[\sum_{j\in[\ell]} \frac{\delta}{\delta m(d)}\mathcal{R}_j(\pi_j^{\text{PL}})\right] \\ &= \mathbb{E}_{\gamma\sim\mu}\left[\sum_{j\in[\ell]} \mathbb{E}_{\sigma_j|\gamma\sim\pi_j^{\text{PL}}}\left[PR^{(j)}_{\sigma,d} + e^{m(d)}\left(\rho_d DR^{(j)}_{\sigma,d} - RI^{(j)}_{\sigma,d}\right)\right]\right].\end{aligned}$$

By linearity of expectation,

$$\begin{aligned}\frac{\delta}{\delta m(d)}\mathcal{R}^{\text{fair}}(\pi^{\text{fair}}) &= \sum_{j\in[\ell]} \mathbb{E}_{\gamma\sim\mu}\left[\mathbb{E}_{\sigma_j|\gamma\sim\pi_j^{\text{PL}}}\left[PR^{(j)}_{\sigma,d} + e^{m(d)}\left(\rho_d DR^{(j)}_{\sigma,d} - RI^{(j)}_{\sigma,d}\right)\right]\right] \\ &= \sum_{j\in[\ell]} \mathbb{E}_{\gamma,\sigma_j\sim\pi^{\text{fair}}}\left[PR^{(j)}_{\sigma,d} + e^{m(d)}\left(\rho_d DR^{(j)}_{\sigma,d} - RI^{(j)}_{\sigma,d}\right)\right]. \quad (10)\end{aligned}$$

Hence, we can estimate each term in Equation 10 by taking an empirical average of M samples of each group-wise rankings. For this, we can take M samples of γ and 1 sample each of σ_j. From [19] we know that for group j we can compute the corresponding term in the summation in time $O(|\mathcal{I}_j| + k \log |\mathcal{I}_j|)$, resulting in a total time complexity of $O(Mk^2\ell + M(|\mathcal{I}| + k\ell \log |\mathcal{I}|))$. Note that this is same as replacing $N = 1$ in Equation 8. ∎

Appendix C. Missing Details of Experiments

Datasets. We perform experiments on datasets, for which several past works have raised fairness concerns and demonstrated the performance of their fair ranking algorithms [8, 19, 24, 27]. The *German Credit* dataset encodes users' creditworthiness as a $0/1$ label [16]. To put this data into query-document pairs, we followed preprocessing similar to [24]. The *MovieLens* dataset consists of user ratings of movies from the movielens.org website [15]. We first performed a singular value decomposition to generate 50-dimensional features. We then chose the largest 5 genres (see Table 1) and kept users that rated at least 50 movies. The *HMDA* dataset consists of data regarding home mortgage loans in the US [11]. We used the preprocessed dataset released by [8]. The HMDA dataset is available for every year since 2007, for all 50 US states. We used the data for Alaska (*AK*) from 2017 and created a train and test split. For a more rigorous testing of our algorithms, we also used Connecticut's (*CT*) data, using years $2013 - 2016$ as training data and year 2017 as test data. We did a PCA pre-processing to reduce feature dimension to 50 [10] and created query-document pairs similar to the German Credit pre-processing in [24]. The details of the datasets are in Table 1.

Datasets with implicit bias. For each dataset, we inject multiplicative implicit bias in the relevance scores of the items from the minority group as a stress test for ranking algorithms. In the HMDA dataset, we multiply the relevance scores of the *female* candidates by β, where β is varied between 0 and 1 across the columns of Figure 3. For datasets with more than two groups, such as *MovieLens*, we use different values of bias for different groups. We report the bias values for all the groups other than *Action* group. This model of bias is inspired by [7], a practical model that gives useful insights about the correct optimization objective to consider.

Hyperparameters. We use a two-layered neural network of 32 hidden units each to predict relevance scores. We use stochastic gradient descent with a learning rate of 0.001 and batch size 512 to optimize our relevance metric. We report aggregate results for 10 runs of each algorithm. We selected other hyperparameters after searching for δ in the range 0.01 to 0.1, M in the range 10 to 100, and k in the range 10 to 30. We chose the final values to be the smallest in the range where implicit bias had a significant impact on the output.

Implementation. The unconstrained PL model was trained using PL-Rank-3 algorithm from [19]. All the experiments were run on an Intel(R) Xeon(R) Silver 4110 CPU (8 cores, 2.1 GHz clock speed, and 128GB DRAM).

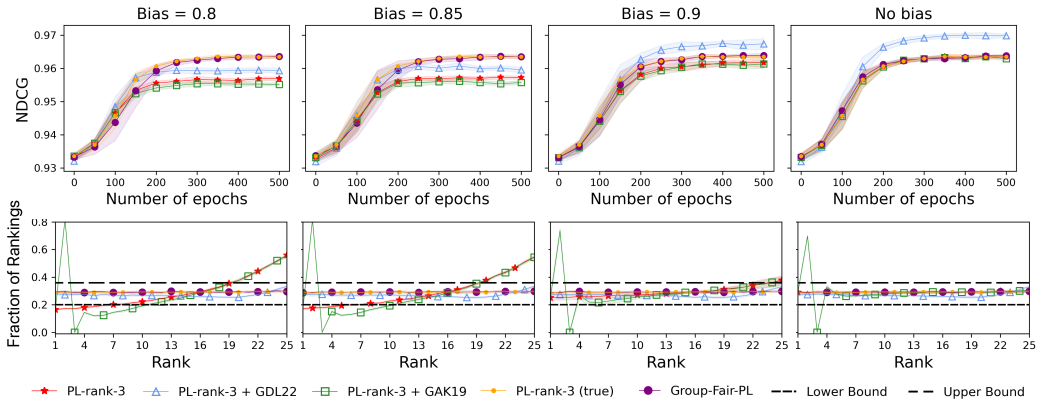

Figure 3: Results on the HMDA (AK) dataset.

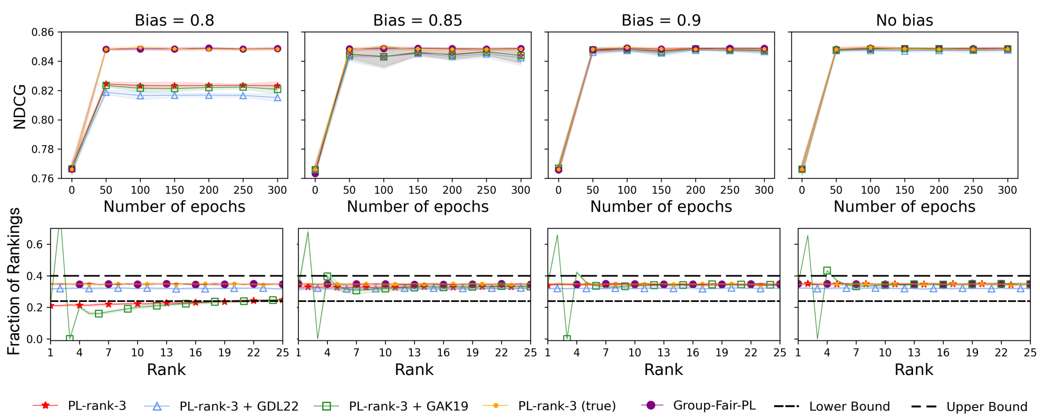

Figure 4: Results on the HMDA (CT) dataset.