A Common Approach for Consumer and Provider Fairness in Recommendations

Dimitris Sacharidis
E-Commerce Research Division
TU Wien, Austria
dimitris@ec.tuwien.ac.at

Kyriakos Mouratidis
School of Information Systems
Singapore Management University, Singapore
kyriakos@smu.edu.sg

Dimitrios Kleftogiannis
Genome Institute of Singapore
A*STAR Research, Singapore
Dimitrios_Kleftogiannis@gis.a-star.edu.sg

ABSTRACT

We present a common approach for handling consumer and provider fairness in recommendations. Our solution requires defining two key components, a classification of items and a target distribution, which together define the case of perfect fairness. This formulation allows distinct fairness concepts to be specified in a common framework. We further propose a novel reranking algorithm that optimizes for a desired trade-off between utility and fairness of a recommendation list.

KEYWORDS

fairness; recommender systems

INTRODUCTION

Fairness in algorithmic decisions means that the system should not discriminate against individuals. For recommender systems [2], fairness may concern either the *consumers* that receive recommendations (C-fairness) — as in [5, 8, 12] — or the *producers/providers* of items being recommended

(P-fairness) — as in [3, 6, 7, 10]. In C-fairness, the issue with algorithmic bias is that users could receive different recommendations based on their protected/sensitive attributes, e.g., age, gender, race. In P-fairness, the issue is that different classes of ranked items (like male vs. female job applicants) could receive unequal attention, or attention disproportional to their overall size in the pool of alternatives.

In this paper, we present a simple unified viewpoint of these fairness concepts, where one only needs to specify two components: (1) an *item classification*, and (2) a *target distribution* over item classes. The exact definition of these components depends on the specific fairness goal one aims for:

• Consider the C-fairness concept where each user should receive *calibrated recommendations* that match her prior preferences, as in [10]; e.g., a user that 70% of the time interacts with items from a particular class, should also receive recommendations that come from that class 70% of the time. Here, the item classification is explicitly given — e.g., movie genres in [10]. The target distribution is specific to each user and represents the proportion of each item class in the user's feedback history.

• In another C-fairness concept, the goal is to provide fair recommendations to a group of users, as in [5, 8]. Here, we introduce an item class per group member: an item belongs to the i-th class if it is within the top-N items of the i-th group member. The notion of top-N-envy-freeness [8] is then captured by setting the target distribution to be uniform over these item classes. Intuitively, we seek to construct a group recommendation list that contain items equally desirable among group members, thus ensuring that no member is envious of the others.

• As an example of P-fairness, consider the scenario where we want to make recommendations that equally represent all item providers, as in [6]. Here, the item classes are the providers. The target distribution is again the uniform distribution over these item classes.

These examples illustrate the flexibility of our approach. Distinct notions of fairness can be described in terms of a simple framework, consisting of an item classification and a target distribution.

Once the two components are defined, the *fairness* of the recommender is measured by the statistical divergence between the target and the *realized* distribution — the latter being the distribution over item classes induced by a recommendation list. The divergence essentially quantifies how far from being perfectly fair the recommendations are. As the goal of a fairness-aware recommender system is to strike a balance between *utility* and *fairness*, similar to prior work [3, 6, 10], we aim to maximize a linear combination of the two measures in a post-processing step. For this purpose, we introduce a novel reranking algorithm that approximately solves this optimization problem.

Overall, our approach allows us to frame various fairness objectives under a common framework, making it possible to relate and compare them. Moreover, our unified algorithm enables the direct optimization of the desired utility-fairness trade-off. This is particularly attractive as some fairness proposals only consider some proxy of fairness.

APPROACH

Item Classification. We assume that items belong to a set of K appropriately defined *classes*; in case of P-fairness, these are typically the item producers/providers, while in C-fairness, classes are specified according to consumer/user properties. We denote as $p(\kappa|t)$ the *membership* of item t in class $\kappa \in \mathcal{K}$. As an item may belong to different classes, we require $\sum_{\kappa \in \mathcal{K}} p(\kappa|t) = 1$.

Target Distribution. A *target distribution* Q is a discrete probability distribution over $\mathcal{K}$, representing the desired class representation a recommendation list should exhibit. For example, we may set Q to the uniform distribution directing the recommender to cover all classes; or to the class distribution of the user's consumed items, asking the recommender to respect past preferences.

Realized Distribution. A recommended list τ presents items from possibly different classes to the target user. The *realized distribution* $P(\tau)$ in a list τ is a discrete probability distribution over the set of classes $\mathcal{K}$, capturing the level at which each class is presented to the user via τ. There are various ways to concretely define $P(\tau)$. The simplest is to sum up the class memberships of each item. Another approach is to account for the *position bias* in a ranked list, and compute the exposure/attention [1, 9] that each class receives in list τ. Briefly, the intuition is that items ranked at the top of the list are more likely to be selected by the user. Specifically, if $d(i)$ denotes the discount for the bias of position i, then a possible definition of the probability of class $\kappa \in \mathcal{K}$ being represented in list τ is:

$$P(\tau)[\kappa] = \gamma_\tau \cdot \sum_{t \in \tau} d(\mathrm{rank}_\tau(t)) \cdot p(\kappa|t) \tag{1}$$

where $\gamma_\tau = \sum_{\kappa \in \mathcal{K}} \sum_{t \in \tau} d(\mathrm{rank}_\tau(t)) \cdot p(\kappa|t)$ ensures that $P(\tau)$ is a probability distribution.

Fairness. A recommended list τ is perfectly fair if its realized distribution is equal to the target. As perfect fairness may be unattainable, we would like to quantify deviations from the ideal state. The *fairness* of a list τ is captured by the similarity of its representation $P(\tau)$ to some given target Q, and is denoted as $F(\tau; Q)$. When the target Q is understood by the context, we simply denote it as $F(\tau)$. Dissimilarity between two distributions can be measured in terms of their statistical divergence. Therefore, we express fairness of τ by a value $F(\tau)$, normalized to $[0, 1]$ as follows:

$$F(\tau) = 1 - \frac{D(P(\tau)\|Q)}{D^+(\cdot\|Q)}, \tag{2}$$

where $D(\cdot\|\cdot)$ is a divergence measure of representation $P(\tau)$ from the target Q, and $D^+(\cdot\|Q)$ is an upper bound on the divergence of any possible representation. When the representation $P(\tau)$ equals the target distribution Q, fairness takes its maximum value $F(\tau) = 1$. In this work, we consider the Kullback–Leibler (KL) divergence, as in [10, 11]: $D(P\|Q) = \sum_{\kappa \in \mathcal{K}} Q[\kappa] \cdot \log\left(\frac{Q[\kappa]}{\tilde{P}[\kappa]}\right)$, where $\tilde{P} =$

$(1 - \alpha) \cdot P + \alpha \cdot Q$ is the smoothed distribution of the class representation for some small constant α (set to 0.01 to match [10]), and $\log(1/\alpha)$ is an upper bound.

Utility. A list has a (predicted) *utility* to the target user, captured by a value $U(\tau)$, normalized to $[0, 1]$. As a concrete example, assume that the recommender estimates the relevance $r(t)$ of an item t to the target user. Let $\mathrm{rank}_\tau(t)$ denote the position of item t in list τ, and let $d(i)$ denote a discount associated with the i-th position in a list — e.g., $d(i) = 1/\log(i + 1)$ as per the discounted cumulative gain (DCG) metric. Then a possible definition of utility is:

$$U(\tau) = \frac{1}{U^+} \sum_{t \in \tau} r(t) \cdot d(\mathrm{rank}_\tau(t)), \tag{3}$$

where U^+ is the maximum possible utility of any ranked list of N items (similar to the ideal DCG).

Optimization Objective. The goal is to return a recommendation list τ that maximizes a linear combination of its utility $U(\tau)$ and its fairness $F(\tau)$, for a given value of $\lambda \in [0, 1]$, i.e.,

$$\arg \max_\tau (1 - \lambda) \cdot U(\tau) + \lambda \cdot F(\tau).$$

Algorithm. The unified algorithm, termed PREFIX, reranks a recommendation list τ_N of length N that maximizes utility alone. Let T denote the set of items to be reranked. The goal is to create a fairness aware reranked list that approximately solves the aforementioned optimization problem. For any list τ, let $S(\tau) = (1 - \lambda) \cdot U(\tau) + \lambda \cdot F(\tau)$ denote its *score*. The algorithm examines each *prefix* τ_k of the initial list, i.e., a list that contains the first k items from τ_N for $k \in [0, N]$. For each prefix τ_k, the algorithm will incrementally append the remaining $N - k$ items in a greedy manner. Let $\tau_k(i)$ denote the list at the i-th step — the final reranked list created from the k-prefix is obtained at the $(N - k)$-th step. At the i-th step, the algorithm appends the item that maximizes the score of the resulting list, i.e., $\arg \max_{t \in T \notin \tau_{i-1}} S(\tau_{i-1} \sqcup \{t\})$, where $\sqcup$ denotes the append operation. After considering all prefixes the algorithm has constructed N, possibly distinct, rerankings of the initial list t_N, and among them, it returns the one with the highest score.
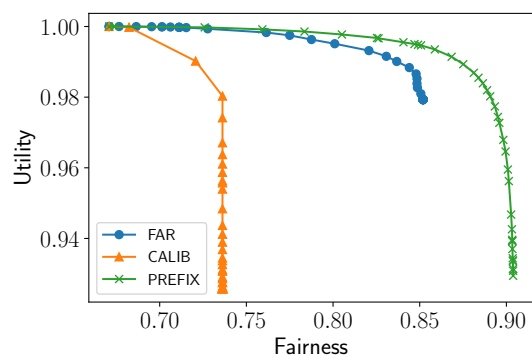
EVALUATION

Data. We use the MovieLens 20M dataset [4]. We create an item class for each movie genre, and for each movie we assign equal membership probabilities to its classes, as in [10]. As typical, we use a basic matrix factorization model with 20 dimensions to create recommendations.
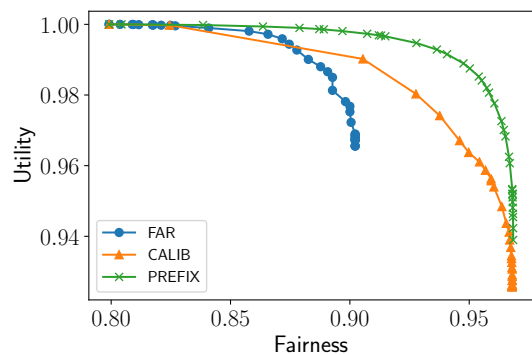
Fairness Objectives and Methods. We consider two scenarios for fairness. *Uniform*, mandates that all the realized distribution is uniform, similar to the P-fairness goal of the FAR method [6]. *User*, requires that the realized distribution follows each user's target distribution from her feedback history, as in the C-fairness goal of the CALIB method [10].

Table 1: Some Utility-Fairness tradeoffs

Uniform Fairness Scenario

	λ	U	R	N	F	C
FAR	0	1.000	0.110	0.110	0.671	0.675
	0.5	0.990	0.091	0.087	0.837	0.883
	1	0.979	0.091	0.083	0.852	**0.898**
CALIB	0	1.000	0.110	0.110	0.671	0.675
	0.5	0.933	0.089	0.072	0.673	0.680
	1	0.926	0.090	0.075	0.674	0.683
PREFIX	0	1.000	0.110	0.110	0.671	0.675
	0.5	0.980	0.047	0.073	0.891	0.898
	1	0.929	0.047	0.061	**0.904**	**0.898**

User Fairness Scenario

	λ	U	R	N	F	D
FAR	0	1.000	0.110	0.110	0.799	1.335
	0.5	0.988	0.111	0.109	0.888	0.746
	1	0.966	0.113	0.082	0.901	0.657
CALIB	0	1.000	0.110	0.110	0.799	1.335
	0.5	0.933	0.089	0.072	0.968	0.214
	1	0.926	0.090	0.075	0.968	0.213
PREFIX	0	1.000	0.110	0.110	0.799	1.335
	0.5	0.982	0.092	0.100	0.957	0.283
	1	0.939	0.093	0.080	**0.968**	**0.210**

Metrics. We ask each method to return a ranked list of 20 items. For a list, we measure: U: the utility according to Equation 3; R: recall at position 20, i.e., recall@20; N: normalized DCG at position 20, i.e., nDCG@20; F: fairness according to Equation 2; C: class coverage ratio for Uniform as in [6]; D: KL divergence for User as in [10]. In all measures except the last, higher means better.

Results. All three methods use a parameter λ to control the tradeoff between utility and fairness. Table 1 shows all metrics, as we increase λ from 0 to 0.5 and 1. For both scenarios and $\lambda = 0$, all methods return the same list. For Uniform and $\lambda = 1$, CALIB performs poorly in this scenario, while PREFIX matches the effectiveness of FAR in terms of coverage (C), and does better in fairness (F) achieving more class-balanced recommendations. For User and $\lambda = 1$, FAR performs poorly, while PREFIX exceeds the effectiveness of CALIB in terms of fairness (F) and in KL divergence (D).

We next set various values to λ within $[0, 1]$ and plot the utility-fairness curve of each generated ranked list, shown in Figure 1. A recommender with a curve going closer to the upper right corner (of perfect utility and fairness) is more desirable, as it better trades off utility against fairness. In the Uniform scenario, it is clear that PREFIX is more effective than FAR and CALIB. In the User scenario, where the highest possible fairness values for PREFIX and CALIB are comparable (rightmost points), observe that the curve of PREFIX again goes closer to the perfect utility-fairness corner.

(a) Uniform

(b) User

Figure 1: Utility-Fairness curves

REFERENCES

[1] Asia J. Biega, Krishna P. Gummadi, and Gerhard Weikum. 2018. Equity of Attention: Amortizing Individual Fairness in Rankings. In *ACM SIGIR*. ACM, 405–414. https://doi.org/10.1145/3209978.3210063

[2] Robin Burke. 2017. Multisided Fairness for Recommendation. *CoRR* abs/1707.00093 (2017). arXiv:1707.00093

[3] Robin Burke, Nasim Sonboli, and Aldo Ordonez-Gauger. 2018. Balanced Neighborhoods for Multi-sided Fairness in Recommendation. In *FAT (Proceedings of Machine Learning Research)*, Vol. 81. PMLR, 202–214.

[4] F. Maxwell Harper and Joseph A. Konstan. 2016. The MovieLens Datasets: History and Context. *TiiS* 5, 4 (2016), 19:1–19:19. https://doi.org/10.1145/2827872

[5] Xiao Lin, Min Zhang, Yongfeng Zhang, Zhaoquan Gu, Yiqun Liu, and Shaoping Ma. 2017. Fairness-Aware Group Recommendation with Pareto-Efficiency. In *ACM RecSys*. ACM, 107–115. https://doi.org/10.1145/3109859.3109887

[6] Weiwen Liu and Robin Burke. 2018. Personalizing Fairness-aware Re-ranking. *CoRR* abs/1809.02921 (2018). arXiv:1809.02921

[7] Tien T. Nguyen, Pik-Mai Hui, F. Maxwell Harper, Loren G. Terveen, and Joseph A. Konstan. 2014. Exploring the filter bubble: the effect of using recommender systems on content diversity. In *WWW*. ACM. https://doi.org/10.1145/2566486.2568012

[8] Dimitris Serbos, Shuyao Qi, Nikos Mamoulis, Evaggelia Pitoura, and Panayiotis Tsaparas. 2017. Fairness in Package-to-Group Recommendations. In *WWW*. ACM, 371–379. https://doi.org/10.1145/3038912.3052612

[9] Ashudeep Singh and Thorsten Joachims. 2018. Fairness of Exposure in Rankings. In *ACM KDD*. ACM, 2219–2228. https://doi.org/10.1145/3219819.3220088

[10] Harald Steck. 2018. Calibrated recommendations. In *ACM RecSys*. ACM, 154–162. https://doi.org/10.1145/3240323.3240372

[11] Ke Yang and Julia Stoyanovich. 2017. Measuring Fairness in Ranked Outputs. In *SSDBM*. https://doi.org/10.1145/3085504.3085526

[12] Sirui Yao and Bert Huang. 2017. Beyond Parity: Fairness Objectives for Collaborative Filtering. In *NIPS*. 2925–2934.