Addressing the Target Customer Distortion Problem in Recommender Systems

Xing Zhao, Ziwei Zhu, Majid Alfifi, and James Caverlee
Department of Computer Science and Engineering, Texas A&M University
xingzhao,zhuziwei,alfifi,caverlee@tamu.edu

ABSTRACT

Predicting the potential target customers for a product is essential. However, traditional recommender systems typically aim to optimize an engagement metric without considering the overall distribution of target customers, thereby leading to serious distortion problems. In this paper, we conduct a data-driven study to reveal several distortions that arise from conventional recommenders. Toward overcoming these issues, we propose a target customer re-ranking algorithm to adjust the population distribution and composition in the Top-k target customers of an item while maintaining recommendation quality. By applying this proposed algorithm onto a real-world dataset, we find the proposed method can effectively make the class distribution of items' target customers close to the desired distribution, thereby mitigating distortion.

CCS CONCEPTS

• **Information systems → Recommender systems**.

KEYWORDS

Recommendation System, Distribution Distortion, Calibration

ACM Reference Format:
Xing Zhao, Ziwei Zhu, Majid Alfifi, and James Caverlee. 2020. Addressing the Target Customer Distortion Problem in Recommender Systems. In *Proceedings of The Web Conference 2020 (WWW '20), April 20–24, 2020, Taipei, Taiwan.* ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3366423.3380065

1 INTRODUCTION

Predicting the potential target customers for a product is essential. Accurate analysis and prediction of the population distribution of target customers for an item (e.g., a product or an advertisement) could directly improve the item's business prospects. One key factor in targeting is the *class distribution* of target customers. For example, an advertising campaign may wish to guarantee at least a certain demographic (e.g., 18-35) sees its ads. Or a job posting may wish to guarantee that an equal number of women and men are targeted.

This focus on the overall distribution and composition of an item's target customers is often in opposition to the criteria driving *personalized* recommenders, which typically aim to optimize an engagement metric without considering the overall distribution of target customers. For example, recommendation algorithms such as Probabilistic Matrix Factorization (PMF) [19], Bayesian Personalized Ranking (BPR) [20], and Neural Collaborative Filtering (NCF) [11], are designed to optimize for metrics based on user-item interactions like mean average precision, NDCG, *precision@k* or *recall@k*. However, these approaches typically do not consider the distribution of target customers.

To illustrate, consider a media-service provider that recommends movies to their users. From the perspective of each movie, what customers will be considered first as the targets to receive the recommendation? One intuitive answer is that the target customers should be the ones who are most potentially interested in this item. There are many approaches to recommend items to users based on past user-item interactions. Suppose at the end, based on the predicted recommendation results, we have a target customer set that receives the recommendation of item i. We use q as the class (such as gender, age, or occupation) distribution of target customers for item i. By relying on a recommender that does not consider the distribution of target customers, consider the following distortions that may arise (and that we verify do arise in our data-driven analysis in this paper):

Distortion 1: the target distribution q may be dominated by the overall class distribution. For example, suppose the male-female ratio of users in the entire customer base is 70%:30%. For each movie, the set of recommended users always contains more males than females, even though some movies may be individually preferred by females.

Distortion 2: the target distribution q for minority classes may be under-recommended for majority-preferred items. For example, suppose movie i is a male-preferred movie, and the male-female ratio of users who have already watched movie i is 90%:10%. There will be fewer females (even much less than 10%) in i's predicted target customers.

Distortion 3: the target distribution q may unexpectedly differ from the appropriate distribution. For example, if i is an R-rated movie, the predicted target customer set may include a great number of children using conventional recommenders.

In this paper, we focus on these types of *target customer distortion problems*, where the distribution and composition of target customers differ much from the desired ones, that arise in recommenders. Recent research has examined related issues in the distribution challenge from the user's perspective through *calibrated recommendations*, to ensure that users are exposed to a diverse recommended list of items [22]. However, there is a gap in viewing this distribution challenge from the item's perspective. That is, what is the distribution of customers that are targeted for each item?

Concretely, we first introduce an approach to identify the target customers for each item (Section 3), which is challenging since recommenders are typically structured to reveal what each user prefers

WWW '20, April 20–24, 2020, Taipei, Taiwan

ACM ISBN 978-1-4503-7023-3/20/04.
https://doi.org/10.1145/3366423.3380065

(rather than what users are targeted by each item). We conduct a data-driven study in Section 4 to reveal several distortions that arise from conventional recommenders. Toward overcoming these issues, in Section 5, we propose a target customer re-ranking algorithm to adjust the population distribution and composition in the Top-k target customers of an item while maintaining recommendation quality. Next, in Section 6, we apply this proposed algorithm onto the MovieLens 1M dataset [10], and evaluate the distribution of users and the quality of recommendation, then discuss its merits and drawbacks. Last but not least, we conclude our work and point out future research opportunities in Section 7.

2 RELATED WORK

There are many studies in using calibration for dealing with distributional issues that arise in classification settings, e.g., [7, 23, 27]. Recently, Steck proposed calibrated recommendation, which focuses on the diversity of genres in a user's recommendation list [22]. Soon after, Kaya and Bridge compared Steck's work with intent-aware recommendations, and proposed a new version of calibration and three new evaluation metrics [15]. Liu *et al.* proposed a Fairness-Aware Re-ranking (FAR) algorithm to balance ranking quality and borrower-side fairness in microlending, to give borrowers from different demographic groups a fair chance of being recommended [18]. These post-processing approaches are all designed from the viewpoint of users. And most of these efforts pay special attention to if the recommended items to a user are fair or diverse, rather than focusing on the overall distribution. Also there are many works studying the related topics of fairness [3, 13, 16, 25, 26, 28] and diversity [1, 4, 8, 9, 12, 29] on recommendation.

3 TARGET CUSTOMERS OF ITEMS

To address the target customer distortion problem, we first need to obtain the target customers of an item predicted by a conventional recommender system. With these target customers, we could then identify the distortions that occur.

Given a user set $\mathcal{U}$, an item set $\mathcal{I}$, and a binary *user-item interaction matrix* $\mathbf{H} \in \mathbb{N}^{|\mathcal{U}|\times|\mathcal{I}|}$ (where $\mathbf{H}_{u,i} = 1$ indicates that user $u \in \mathcal{U}$ has watched movie $i \in \mathcal{I}$ for example), traditional recommenders output predicted recommendation results which we represent as a *score matrix* $\mathbf{D} \in \mathbb{R}^{|\mathcal{U}|\times|\mathcal{I}|}$. Each value $\mathbf{D}_{i,j}$ expresses the predicted score from user $u \in \mathcal{U}$ to item $i \in \mathcal{I}$.

To obtain the Top-k recommended items for user u, we can return the first k items with the largest predicted score in row $\mathbf{D}_{u,\_}$, calculated as follows ($\searrow$ symbolizes descending sort):

$$top^{\mathcal{U}}(u, k) = \underset{\searrow, k}{\arg\,\mathrm{sort}} \langle \mathbf{D}_{u,1}, \mathbf{D}_{u,2}, \ldots, \mathbf{D}_{u,|I|}, \rangle \tag{1}$$

A straightforward idea is using the same way to obtain the Top-k predicted target customers of an item, directly from the score matrix $\mathbf{D}$. However, by leveraging the conventional personalized recommenders, e.g., [2, 6, 11, 20, 21, 24], we cannot directly compare the predicted scores in a column (for one item) in the score matrix, due to the different users' bias. That is, for example, a higher predicted score for user-item pair (u_1, i) does not necessarily indicate that u_1 likes item i more than another user u_2 who has a lower predicted score for i, because u_1 and u_2 may have different scoring scales.

Therefore, to obtain the Top-k predicted target customers of an item, we should first normalize $\mathbf{D}$ column-wise. Instead of using the predicted value, we normalize $\mathbf{D}$ using the rank information from a user to an item. We map each value $\mathbf{D}_{u,i}$ to its **descending rank order** in the row $\mathbf{D}_{u,\_}$, and define a *ranking matrix* $\mathbf{R} \in \mathbb{N}^{|\mathcal{U}|\times|\mathcal{I}|}$ mapping scores to ranks from $\mathbf{D}$. For example, if a row in $\mathbf{D}$ is $\langle 0.9, 0.3, 0.6\rangle$, the mapped row in $\mathbf{R}$ should be $\langle 1, 3, 2\rangle$. Values of $\mathbf{R}$ are integers between 1 to $|\mathcal{I}|$.

On the one hand, from the user's perspective, each value of the ranking matrix, $\mathbf{R}_{u,i}$ represents u's preference rank to i among all items. Therefore, the Top-k recommended items for a user u could also be represented as follows ($\nearrow$ symbolizes ascending sort):

$$top^{\mathcal{U}}(u, k) = \underset{\nearrow, k}{\arg\,\mathrm{sort}} \langle \mathbf{R}_{u,1}, \mathbf{R}_{u,2}, \ldots, \mathbf{R}_{u,|I|} \rangle \tag{2}$$

On the other hand, from the item's perspective, each value of the ranking matrix, $\mathbf{R}_{u,i}$, represents the preference degree from user u to this item i among all users. Target customers are the users who are most potentially interested in an item (with highest preference degree). Now, the Top-k target customers for an item i could be represented as:

$$top^{\mathcal{I}}(i, k) = \underset{\nearrow, k}{\arg\,\mathrm{sort}} \langle \mathbf{R}_{1,i}, \mathbf{R}_{2,i}, \ldots, \mathbf{R}_{|U|,i} \rangle \tag{3}$$

Furthermore, we know each row, $\mathbf{R}_{u,\_}$, represents the preference ranking given from a user u to each item $i \in \mathcal{I}$. Suppose values in the originally predicted recommendation matrix $\mathbf{D}$ are different from one another, and the values in $\mathbf{R}_{u,\_}$ must be from 1 to $|\mathcal{I}|$. However, if we vertically observe $\mathbf{R}$, we can find the values in column $\mathbf{R}_{\_,i}$ may be identical to one another, and the value range is **not** necessarily from 1 to user size $|\mathcal{U}|$. For instance, assuming there is a trendy item i_{hot} which has been set as many users' first priority, so that the column $\mathbf{R}_{\_,i_{hot}}$ should contain many "1"s and the average (or median) of $\mathbf{R}_{\_,i_{hot}}$ could be much less than a column for an unpopular item. Keeping this observation in mind, we will focus not only on all item's class distribution but also on popular items.

4 DATA-DRIVEN STUDY

In this section, we focus on three examples from a real-world dataset, and observe the distortion between the desired class distribution (denoted as p) and the class distribution of predicted Top-k target customers (denoted as q) for an item.

4.1 Dataset

We adopt the MovieLens 1M dataset [10], which contains 1 million user-movie interactions collected from 6,040 users ($\mathcal{U}$) and 3,706 movies ($\mathcal{I}$). We only consider user-item interactions rather than the explicit ratings, i.e. all interacted user-movie pairs will be considered as 1. In addition, for each user, this dataset contains user profile information, including age, gender, and occupation, which could help us to analyze the class distribution of the audience. We now start from gender and age as the concerned demographic classes in the beginning to identify the three distortions from Section 1.

For clarity in presentation, we adopt Bayesian Personalized Ranking (BPR) [20], one of the most influential and foundational personalized recommender algorithms. Experiments with other tested

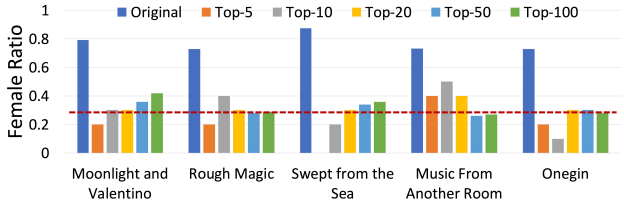


Figure (1) The female ratio on some female-preferred movies. The dotted red line is the female ratio of the entire dataset.

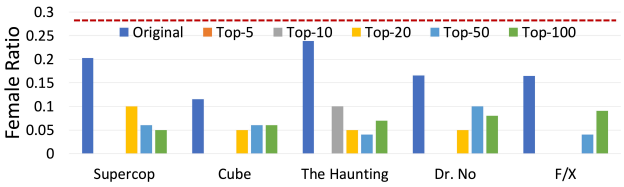


Figure (2) The female ratio on some male-preferred movies. The dotted red line is the female ratio of the entire dataset.

algorithms show similar results; our emphasis here is on the general problem of target customer distortion that can manifest in recommenders that optimize for engagement metrics without consideration of the overall item target distribution. We apply BPR and obtain the predicted user-item interaction matrix **D**. Through the transformations introduced in Section 3, we find the predicted target customers for each item. In the following analysis, we would like to observe the class distribution of the Top-k target customers, $top^I(i,k)$, of certain items. Intuitively, we expect the predicted class distribution q of target customers to match the distribution p of existing users in the training set of an item. To this end, we sort all users who interacted with an item by their timestamps. Next, for each item, we select the first 60% interactions as the training set and randomly select half of the rest data as validation set (20%), then, use the remaining 20% as the test dataset.

4.2 Examples of Target Customer Distortion

Intuitively, we expect that the desired class distribution p and the predicted class distribution q should be as similar as possible. However, following the three distortions in Section 1 we find there is a strong disagreement between the two distributions in many cases:

Case 1: Fewer females are the target customers of female-preferred movies. Figure 1 shows the gender distribution of 5 female-preferred movies in their training set (p, blue bar), and corresponding distribution in their Top-k predicted target customer set (q, bars with other colors). We observe that even though in the training set, females have much more interactions than males with these movies, the predicted target customers still contain more males than females, and for all settings of the Top-5, Top-10, Top-20, Top-50, and Top-100 target customers. One of the reasons is due to the overall smaller ratio (0.29) of females in the entire training set (refer to the dotted red line). In this example of distortion, the recommender under-serves a large target customer group (in this case, females).

Case 2: Few females are the target customers of male-preferred movies. Figure 2 shows the gender distribution of five male-preferred movies in the training set and Top-k target customer set. The historical training data show more males interacted with these movies than females. In this case, it is expected that fewer females may be

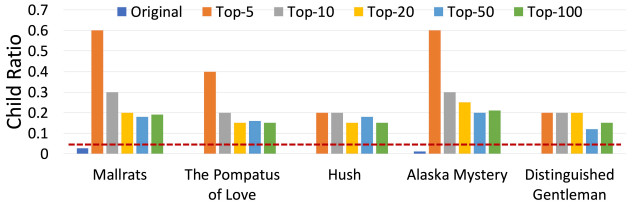


Figure (3) The ratio of children on some R-rated movies. The dotted red line is the child ratio of the entire dataset.

included in the target customers set. But, this does not mean the recommender should ignore females in their Top-K target customer set. In fact, even though females do watch these movies, however, in the predicted result, there are few females in the Top-5 and Top-10. Although this situation is relaxed when we select more candidates as target customers, the ratio of females is still much lower than the desired one. In this example of distortion, the recommender under-serves a small-size class (in this case, female), or sometimes ignores them completely. A similar phenomenon has also been analyzed in Steck's research from the user perspective [22]: some genres with a small portion will be less recommended to a user.

Case 3: Children are target customers of R-rated movies. Figure 3 shows the ratio of children (age under 18) for five R-rated movies. Although the R-rated movie should not be targeted to children under 18-years-old, we still observe some cases in our training set. As shown in Figure 3, the recommender includes a substantial portion of children as target customers of R-rated movies. One of the hypotheses of the phenomena shown in Case 1 and 2 is due to the female ratio (0.29) over the entire dataset being lower than males (0.71). However, the phenomenon of predicting more children as target customers of R-rated movies surprisingly violates the aforementioned hypothesis. That is, in the entire dataset the ratio of children (age under 18) is only around 0.04, but the predicted ratio of children in target customers of R-rated movies are much more than that. From this case, we can observe that in some cases conventional recommenders may over-serve a tiny class (i.e., children).

5 TARGET CUSTOMER RE-RANKING

In Section 4.2, we have shown the predicted class distribution of target customers (q) strongly disagrees with the expected one (p) recommended by using a conventional recommender. Intuitively, one potential solution for these issues is to re-generate the Top-k predicted target customers set to make the class distribution of target customers for an item fit the desired distribution. In many cases, it may be reasonable to set the desired distribution $p(o|i)$ as the class distribution in the training set. Furthermore, there may be some special cases where we wish to manually control the distribution $p(o|i)$, e.g., setting $p(o = child|i_{R-rated}) = 0$ to limit children from being recommended R-rated movies.

Problem Statement: Given a predicted item-user interaction ranking matrix **R** by a conventional recommender, a concerned class o (e.g., gender or age), and the desired class distribution $p(o|i)$, we aim to make the class distribution of predicted target customers, $q(o|i)$, be as similar as $p(o|i)$ through a re-ranking process, while maintaining the original recommendation performance.

As many recommenders are trained in a pairwise manner, many studies state that one might not be able to include calibration into

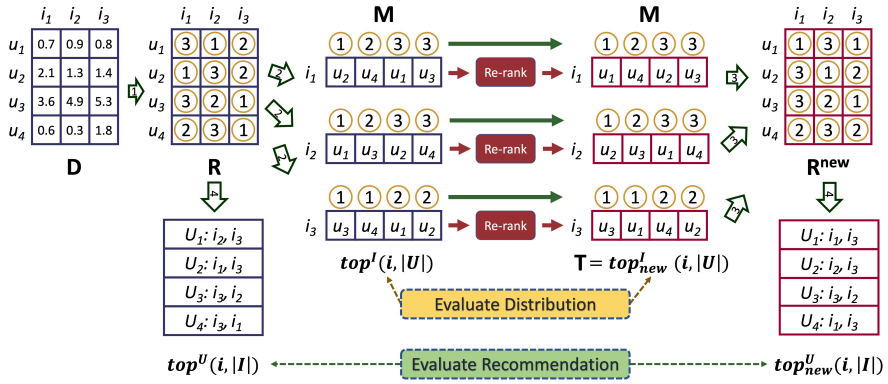


Figure (4) Target customer re-ranking algorithm: (1) from *score matrix* D predicted by a conventional recommender, generate the original *ranking matrix* R; (2) from R generate the Top-k recommended item set $top^{\mathcal{U}}(u,k)$ for each user for later evaluation; (3) from R generate the Top-k target customer set $top^{\mathcal{I}}(i,k)$ for each item as well as the *memory matrix* M; (4) re-rank the Top-k target customer set and generate a new *target customer matrix* $\mathbf{T}_i = top^{\mathcal{I}}_{new}(i,|\mathcal{U}|)$ and evaluate the distribution $KL(p||q)$ through comparing $top^{\mathcal{I}}_{new}(i,k)$ and $top^{\mathcal{I}}(i,k)$; (5) from T and M generate the *new ranking matrix* $\mathbf{R}^{new}$; (6) from $\mathbf{R}^{new}$ generate the new Top-k recommended item set $top^{\mathcal{U}}_{new}(u,k)$; and (7) evaluate the recommendation by comparing $top^{\mathcal{U}}_{new}(u,k)$ and $top^{\mathcal{U}}(u,k)$.

the training [22]. Therefore, a common solution is re-ranking the predicted list in a post-processing step, which has been widely used in calibrated machine learning approaches [7, 22, 27]. In this section, we propose a post-processing approach for target customer re-ranking to make the class distribution of target customers of each item as close as the desired class distribution.

5.1 Class Distribution

We have introduced how to get the Top-k predicted target customers for each item i, $top^{\mathcal{I}}(i,k)$, from the ranking matrix $\mathbf{R}$ (refer to Section 3). From the user set $top^{\mathcal{I}}(i,k)$, we can now analyze the Top-k target customer's class distribution, such as gender and age. Given o as the class of interest, where o could represent gender or age range (or other domain-specific class of interest), we denote for each valid value c for o, the *desired class distribution* $p(o=c|i)$ for item i as:

$$p(o=c|i) = \frac{\sum_{u\in\mathcal{U}} \omega_{u,i} \times p(o=c|u)}{\sum_{u\in\mathcal{U}} \omega_{u,i}} \tag{4}$$

where $p(o=c|u)$ and $\omega_{u,i}$ are two binary variables: $p(o=c|u)$ is 1 if u belongs to c, and $\omega_{u,i}$ is 1 if user u watched movie i in the training dataset, respectively. When o represents gender, we assume given a movie i, the probability $p(o=male|i)$ is the ratio of males to all people who watched this movie, and $p(o=female|i)$ is the ratio of females, supposing for simplicity in presentation here that *male* and *female* are mutually exclusive. For a given historical matrix $\mathbf{H}$, the desired class distribution $p(o|i)$ could be a fixed number (ratio) based on the historic interaction record. In some special occasions, $p(o|i)$ could also be manually set as a desired number, for example, we could set $p(o=child)|i_{R-rated}) = 0$ to expect that all children (age under 18) should not be recommended R-rated movies.

Similarly, we could calculate the *predicted class distribution*, $q(o|i)$, of predicted Top-k target customers $top^{\mathcal{I}}(i,k)$ for item i as follows:

$$q(o|i) = \frac{\sum_{u\in top^{\mathcal{I}}(i,k)} p(o|u)}{k}. \tag{5}$$

Ideally, we expect the *predicted class distribution* $q(o|i)$ to be as similar as the *desired class distribution* $p(o|i)$. Otherwise, if $q(o|i)$ is quite different from $p(o|i)$, then we will have identified a *target customer distortion*. In most of the cases, we expect the desired distribution $p(o|i)$ is the historical distribution of existing users for an item i, as calculated in Eq. 4. We also allow manually setting p in some cases. For example, as in the third case in Section 4, some children (age under 18) watched the R-rated in our training dataset so that the distribution of $p(o=child|i) \geq 0$ for R-rated movie i. Even though, we still could manually force $p(o=child|i) = 0$ to avoid recommending R-rated movie i to children by ranking all children in the very back of the potential relative user list of i.

To compare the similarity/distance between two distributions $p(o|i)$ and $q(o|i)$, we use the Kullback-Leibler (KL) divergence [17] as the metric, where $KL(p||q) = 0$ indicates the distributions $p(o|i)$ and $q(o|i)$ are exactly the same; and a larger $KL(p||q)$ indicates they are opposite of each other.

5.2 KL-weighted Top-k Target Customers

In Section 3, we showed how to get the Top-k predicted target customers $top^{\mathcal{I}}(i,k)$ from the ranking matrix $\mathbf{R}$. To memorize the preference priority from each user $u \in top^{\mathcal{I}}(i,|\mathcal{U}|)$ to item i, we introduce a *memory matrix* $\mathbf{M} \in \mathbb{N}^{|\mathcal{I}|\times|\mathcal{U}|}$:

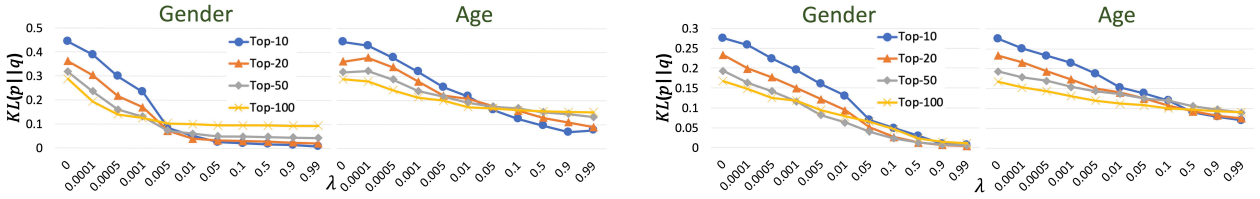
$$\mathbf{M}_i = \underset{\nearrow}{\text{sort}}\langle \mathbf{R}_{1,i}, \mathbf{R}_{2,i}, \ldots, \mathbf{R}_{|\mathcal{U}|,i} \rangle \tag{6}$$

recalling that $\mathbf{R}_{(u,i)}$ is the **rank** of item i in user u's priority list.

To re-rank the Top-k most likely target customers and let the class distribution of target customers q to fit our desired distribution p, we leverage *maximum marginal relevance (MMR)*, which can provide precise re-ranking results [5]. We store these re-ranked results into the new *target customers matrix* $\mathbf{T} \in \mathbb{N}^{|\mathcal{U}|\times|\mathcal{I}|}$, so that the new Top-k predicted target customers for item i, i.e., $top^{\mathcal{I}}_{new}(i,k)$, is the first k elements (users) in i^{th} column of $\mathbf{T}$. We could obtain the optimized new target customer set, $\mathbf{T}_i$, for item i as follows:

$$\mathbf{T}_i = \underset{C_i \subseteq top^{\mathcal{I}}(i,|\mathcal{U}|)}{\arg\max} \quad (1-\lambda) \times r(C_i) - \lambda \times KL(p||q(C_i)) \tag{7}$$

where $\lambda \in [0,1]$ is the trade-off between the original recommendation results and the distribution metric, C is the current optimal subset of re-ranked target customers, and recommendation score



(a) All items **(b) Popular items**

Figure (5) $KL(p||q)$ **of class distributions of target customers after applying the target customer re-ranking algorithm.**

$r(C)$ is calculated from the ranking (priority) of user $u \in C$ in item i's original target customers list:

$$r(C_i) = \frac{1}{|C_i|}\left(\sum_{u \in C_i} \frac{1}{\mathbf{R}_{u,i}+1}\right) \tag{8}$$

Through the re-ranking process, a user's ID can be stored in a column $\mathbf{T}_i$ per step, from top to bottom.

5.3 Top-Z Selection Mechanism

To add each user into $\mathbf{T}_i$, a traditional re-ranking method would go through the entire original target customer list, $top^{\mathcal{I}}(i, |\mathcal{U}|)$ with size of $|\mathcal{U}|$), then select the one with the most optimal KL-weighted score. To save running time and maintain prediction quality, instead of going through the entire list of $top^{\mathcal{I}}(i, |\mathcal{U}|)$, we only consider the Top-Z users in $top^{\mathcal{I}}(i, |\mathcal{U}|)$; in our experiments, we set Z as 30 times the number of valid values for o (e.g., $Z = 60$ for $o \in$ *Gender*).

The benefits of only selecting from Top-Z users in the *current* $top^{\mathcal{I}}(i, Z)$ rather than the entire user set are not only significantly speeding processing time ($Z \ll |\mathcal{U}|$), but also further ensuring recommendation quality. That is, we need not engage our re-ranking algorithm to choose the user in the bottom of $top^{\mathcal{I}}(i, |\mathcal{U}|)$, although it may slightly improve $KL(p||q)$.

5.4 Rebuild the Rank Matrix for Users

In Section 3, we introduced how to transform $\mathbf{D} \rightarrow \mathbf{R} \rightarrow top^{\mathcal{I}}(i, k)$. As every step is a linear transformation, the entire process can be reversed. That means from the re-ranked target customers matrix $\mathbf{T}$ where $\mathbf{T}_i = top^{\mathcal{I}}_{new}(i, |\mathcal{U}|)$, we could reverse the process through $\mathbf{T} \rightarrow \mathbf{R}^{new}$ and generate the new version of the ranking matrix $\mathbf{R}^{new}$. Specifically, leveraging the re-ranked target customer matrix $\mathbf{T}$ and the original memory matrix $\mathbf{M}$, we could build the new ranking matrix $\mathbf{R}^{new}$ by:

$$\mathbf{R}^{new}_{u,i} = \mathbf{M}_{v(\text{where } \mathbf{T}_{i,v}=u),i} \tag{9}$$

In this way, the new Top-K recommended items for a user u could be easily calculated by:

$$top^{\mathcal{U}}_{new}(u, k) = \underset{\nearrow, k}{\arg\text{sort}} \langle \mathbf{R}^{new}_{u,1}, \mathbf{R}^{new}_{u,2}, \ldots, \mathbf{R}^{new}_{u,|I|} \rangle \tag{10}$$

Also, to check the recommendation performance of the re-ranked matrix $\mathbf{R}^{new}$, we leverage the widely-used evaluation metric, $F-1@K$. Through comparing with the original Top-k recommended items to a user (in $\mathbf{R}$) and the new Top-k recommended items for the same user (in $\mathbf{R}^{new}$) after using the proposed target customer re-ranking algorithm, we can measure the impact on the recommendation results after considering the class distribution of target customers. To illustrate the workflow of the proposed re-ranking algorithm, Figure 4 walks step-by-step through a simple example.

6 EXPERIMENTAL RESULTS AND ANALYSIS

In the previous sections, we have identified how the class distribution of target customers of an item (q) and it's desired distribution (p) can be distorted. To address this problem, we proposed a target customer re-ranking algorithm. In this section, we apply the proposed algorithm onto the MovieLens dataset introduced in Section 4, and evaluate the results from both perspectives of distribution bias and recommendation accuracy.

To match the result analysis in Section 4, we use BPR as the base of our target customer re-ranking algorithm. It is important to note that the proposed algorithm is a post-processing solution which could be applied upon any conventional recommenders. Here we use BPR as a representative. We first apply BPR onto our training dataset, and through the transformation introduced in Section 3 we obtain the Top-k predicted target customers $top^{\mathcal{I}}(i, k)$ for each user i. Through applying the proposed re-ranking approach onto $top^{\mathcal{I}}(i, k)$ we now have the re-ranked new Top-k predicted target customers $top^{\mathcal{I}}_{new}(i, k)$ for i. Furthermore, we can also obtain users' Top-k recommended items before and after applying the re-ranking approach, i.e. $top^{\mathcal{U}}(u, k)$ and $top^{\mathcal{U}}_{new}(u, k)$, respectively.

6.1 Bias of Class Distributions of Target Customers

First of all, we compare the desired class distribution p with $top^{\mathcal{I}}(i, k)$ and $top^{\mathcal{I}}_{new}(i, k)$, respectively. Figure 5a shows the score of the distribution metric $KL(p||q)$ with different settings of λ, in two cases, i.e., $o \in$ *Gender* and $o \in$ *Age*, for all items. In both cases, we observe that $KL(p||q)$ decreases with the increase of λ. The difference among these two cases are: comparing with the case of $o \in$ *Gender*, $KL(p||q)$ is harder to converge in the case of $o \in$ *Age*, and the $KL(p||q)$ is still far from 0 when we set the largest λ in our experiment. This is because we choose the optimal user u in the Top-Z candidates of original $top^{\mathcal{I}}(i, k)$ each interaction, instead of the entire user set. Within an extreme condition, Top-Z candidates do not contain an optimal choice to improve the current $KL(p||q)$. Such a phenomenon becomes even more evident when the class contains more valid values, i.e., there are 7 valid values for *Age* and 2 valid values for *Gender*.

Recalling our Top-Z candidates mechanism, in the case of the small size of class (e.g., *Gender*), the Top-Z candidates mechanism performs well due to the fast processing speed, high recommendation accuracy, as well as almost unharmed $KL(p||q)$ value. However,

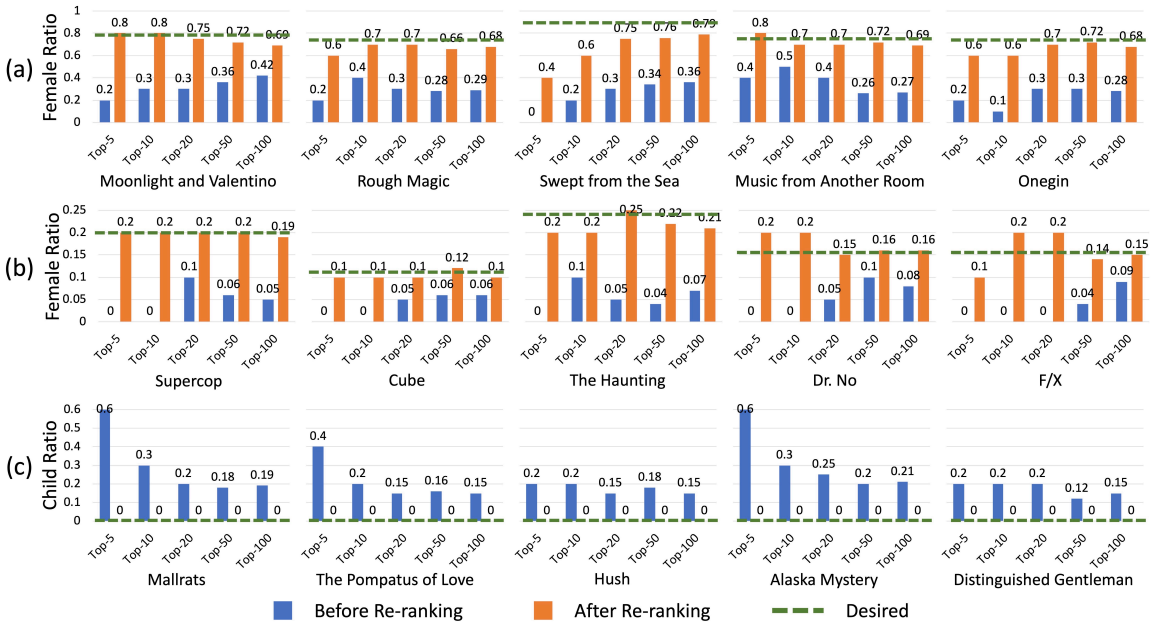


Figure (6) The adjusted ratios of (a) females on sample female-preferred movies, (b) females on sample male-preferred movies, and (c) children on sample R-rated movies before and after re-ranking, in the setting of $\lambda = 0.5$. The dashed lines are the corresponding desired p.

as we can see, with the growing size of class, the effect caused by this selection mechanism to $KL(p||q)$ will be more obvious.

We also observe that, $KL(p||q)$ with a smaller k drops more quickly and converges more slowly with the growth of λ, than $KL(p||q)$ with a larger k. This is due to the original ratio of one class in the entire user set. Given a user set contains 10 females and 90 males and desired distribution $\frac{f}{m} = \frac{1}{2}$, the Top-10 target customers with the expected distribution should obviously be easier to satisfy than the Top-100 target customers.

Figure 5b shows KL scores of *popular items*. Here, we define the popular items as items in users' Top-k preference list. Comparing with the class distribution of all items (refer to Figure 5a), similar downtrends are observed: $KL(p||q)$ will decrease with the growth of λ. However, the $KL(p||q)$ is always lower for popular items than the one for all items, especially when λ is quite small. This observation indicates that a traditional recommender brings more bias of class distributions for unpopular items.

Furthermore, Figure 6 shows the re-ranked Top-k target customers using our re-ranking algorithm of those examples we showed in Section 4, in the setting of $\lambda = 0.5$. It is not surprising that the re-ranked Top-k target customers fit the desired distribution. A reasonable portion of females are included in to corresponding movie's target customer set (refer to Figure 6a and 6b). And no child will be set as target customers of R-rated movie by setting the desired distribution $p(o = children|i_{R-rated}) = 0$ (refer to Figure 6c).

6.2 Influence of Recommendation Accuracy Cased by Re-ranking

There is an inherent trade-off between a reasonable class distribution and an accurate recommendation. We already showed good results of the class distribution of target customers for an item using the proposed re-ranking algorithm. Next, we show how recommendation accuracy is affected.

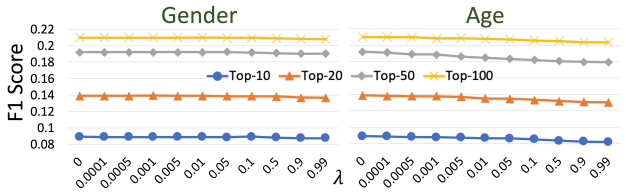


Figure (7) F-1 Score only mildly affected after applying the target customer re-ranking algorithm.

Figure 7 shows the F-1 score of user-viewed Top-k recommendations after applying the proposed target customer re-ranking algorithm optimized for gender and age class distribution, respectively. As we can see, in the case of optimizing gender distribution, with the growth of λ, the recommendation accuracy is almost unaffected. Taking the benefits of class with fewer valid values, the re-ranking is extremely slight, with little impact on F-1. Surprisingly, even in the case of class with more valid values, e.g., *Age*, the recommendation accuracy only drops sightly by taking the benefit of the Top-Z selection mechanism. From an item-viewed perspective, some "ranking metrics" may be affected, e.g., NDCG. However, NDCG is not always the appropriate ranking metric for target customer prediction, because target customers are usually considered as a group, e.g., group ads injection [14].

7 CONCLUSION AND FUTURE WORK

In this paper, we proposed a target customer re-ranking algorithm for addressing the target customer distortion problem. We have seen how the method can effectively adjust the class distribution of the target customers for items toward a desired distribution, thereby mitigating the distortion problem. In future work, we would like to design an advanced re-ranking algorithm which can take care of

multiple classes at the same time. That is, supposing o_i and o_j are not mutually exclusive; for example, considering the gender and age distribution together, how could we adjust the distribution of o_i and o_j simultaneously to make them close to their own desired distributions respectively? We would also like to design an end-to-end recommender system that balances recommendation quality and target customer distributions simultaneously, as a complement to the post-processing step introduced here.

REFERENCES

[1] Azin Ashkan, Branislav Kveton, Shlomo Berkovsky, and Zheng Wen. 2014. Diversified Utility Maximization for Recommendations.. In *RecSys Posters*.

[2] Trapit Bansal, David Belanger, and Andrew McCallum. 2016. Ask the gru: Multi-task learning for deep text recommendations. In *Proceedings of the 10th ACM Conference on Recommender Systems*. ACM, 107–114.

[3] Alex Beutel, Jilin Chen, Tulsee Doshi, Hai Qian, Li Wei, Yi Wu, Lukasz Heldt, Zhe Zhao, Lichan Hong, Ed H Chi, et al. 2019. Fairness in Recommendation Ranking through Pairwise Comparisons. *arXiv preprint arXiv:1903.00780* (2019).

[4] Aditya Bhaskara, Mehrdad Ghadiri, Vahab Mirrokni, and Ola Svensson. 2016. Linear relaxations for finding diverse elements in metric spaces. In *Advances in Neural Information Processing Systems*. 4098–4106.

[5] Jaime G Carbonell and Jade Goldstein. 1998. The use of MMR, diversity-based reranking for reordering documents and producing summaries.. In *SIGIR*, Vol. 98. 335–336.

[6] Paul Covington, Jay Adams, and Emre Sargin. 2016. Deep neural networks for youtube recommendations. In *Proceedings of the 10th ACM conference on recommender systems*. ACM, 191–198.

[7] Dean P Foster and Rakesh V Vohra. 1998. Asymptotic calibration. *Biometrika* 85, 2 (1998), 379–390.

[8] Mike Gartrell, Ulrich Paquet, and Noam Koenigstein. 2016. Bayesian low-rank determinantal point processes. In *Proceedings of the 10th ACM Conference on Recommender Systems*. ACM, 349–356.

[9] Moritz Hardt, Eric Price, Nati Srebro, et al. 2016. Equality of opportunity in supervised learning. In *Advances in neural information processing systems*. 3315–3323.

[10] F Maxwell Harper and Joseph A Konstan. 2016. The movielens datasets: History and context. *Acm transactions on interactive intelligent systems (tiis)* 5, 4 (2016), 19.

[11] Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua. 2017. Neural collaborative filtering. In *Proceedings of the 26th international conference on world wide web*. International World Wide Web Conferences Steering Committee, 173–182.

[12] Neil Hurley and Mi Zhang. 2011. Novelty and diversity in top-n recommendation–analysis and evaluation. *ACM Transactions on Internet Technology (TOIT)* 10, 4 (2011), 14.

[13] Toshihiro Kamishima, Shotaro Akaho, Hideki Asoh, and Jun Sakuma. 2013. Efficiency Improvement of Neutrality-Enhanced Recommendation.. In *Decisions@ RecSys*. Citeseer, 1–8.

[14] Dimitri Kanevsky, Wlodek W Zadrozny, and Alexander Zlatsin. 2009. System and method for group advertisement optimization. US Patent 7,548,874.

[15] Mesut Kaya and Derek Bridge. 2019. A comparison of calibrated and intent-aware recommendations. In *Proceedings of the 13th ACM Conference on Recommender Systems*. ACM, 151–159.

[16] Jon Kleinberg, Sendhil Mullainathan, and Manish Raghavan. 2016. Inherent trade-offs in the fair determination of risk scores. *arXiv preprint arXiv:1609.05807* (2016).

[17] Solomon Kullback and Richard A Leibler. 1951. On information and sufficiency. *The annals of mathematical statistics* 22, 1 (1951), 79–86.

[18] Weiwen Liu, Jun Guo, Nasim Sonboli, Robin Burke, and Shengyu Zhang. 2019. Personalized fairness-aware re-ranking for microlending. In *Proceedings of the 13th ACM Conference on Recommender Systems*. ACM, 467–471.

[19] Andriy Mnih and Ruslan R Salakhutdinov. 2008. Probabilistic matrix factorization. In *Advances in neural information processing systems*. 1257–1264.

[20] Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. 2009. BPR: Bayesian personalized ranking from implicit feedback. In *Proceedings of the twenty-fifth conference on uncertainty in artificial intelligence*. AUAI Press, 452–461.

[21] Suvash Sedhain, Aditya Krishna Menon, Scott Sanner, and Lexing Xie. 2015. Autorec: Autoencoders meet collaborative filtering. In *Proceedings of the 24th International Conference on World Wide Web*. ACM, 111–112.

[22] Harald Steck. 2018. Calibrated recommendations. In *Proceedings of the 12th ACM conference on recommender systems*. ACM, 154–162.

[23] Özge Sürer, Robin Burke, and Edward C Malthouse. 2018. Multistakeholder recommendation with provider constraints. In *Proceedings of the 12th ACM Conference on Recommender Systems*. ACM, 54–62.

[24] Hao Wang, Naiyan Wang, and Dit-Yan Yeung. 2015. Collaborative deep learning for recommender systems. In *Proceedings of the 21th ACM SIGKDD international conference on knowledge discovery and data mining*. ACM, 1235–1244.

[25] Blake Woodworth, Suriya Gunasekar, Mesrob I Ohannessian, and Nathan Srebro. 2017. Learning non-discriminatory predictors. *arXiv preprint arXiv:1702.06081* (2017).

[26] Sirui Yao and Bert Huang. 2017. Beyond parity: Fairness objectives for collaborative filtering. In *Advances in Neural Information Processing Systems*. 2921–2930.

[27] Bianca Zadrozny and Charles Elkan. 2001. Obtaining calibrated probability estimates from decision trees and naive Bayesian classifiers. In *Icml*, Vol. 1. Citeseer, 609–616.

[28] Ziwei Zhu, Xia Hu, and James Caverlee. 2018. Fairness-aware tensor-based recommendation. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*. ACM, 1153–1162.

[29] Cai-Nicolas Ziegler, Sean M McNee, Joseph A Konstan, and Georg Lausen. 2005. Improving recommendation lists through topic diversification. In *Proceedings of the 14th international conference on World Wide Web*. ACM, 22–32.