When Collaborative Filtering is not Collaborative: Unfairness of PCA for Recommendations

David Liu
Northeastern University
Boston, USA
liu.davi@northeastern.edu

Jackie Baek
NYU Stern
New York, USA
baek@stern.nyu.edu

Tina Eliassi-Rad
Northeastern University
Boston, USA
tina@eliassi.org

ABSTRACT

We study the fairness of dimensionality reduction methods for recommendations. We focus on the established method of principal component analysis (PCA), which identifies latent components and produces a low-rank approximation via the leading components while discarding the trailing components. Prior works have defined notions of "fair PCA"; however, these definitions do not answer the following question: what makes PCA *unfair*? We identify two underlying mechanisms of PCA that induce unfairness at the item level. The first negatively impacts less popular items, due to the fact that less popular items rely on trailing latent components to recover their values. The second negatively impacts the highly popular items, since the leading PCA components specialize in individual popular items instead of capturing similarities between items. To address these issues, we develop a polynomial-time algorithm, *Item-Weighted PCA*, a modification of PCA that uses item-specific weights in the objective. On a stylized class of matrices, we prove that *Item-Weighted PCA* using a specific set of weights minimizes a popularity-normalized error metric. Our evaluations on real-world datasets show that *Item-Weighted PCA* not only improves overall recommendation quality by up to 0.1 item-level AUC-ROC but also improves on both popular and less popular items.

1 INTRODUCTION

The growing prevalence of machine learning algorithms across diverse fields motivates the importance of understanding the underlying mechanisms that drive these algorithms' decision-making processes. Within this context, this paper focuses on a specific algorithm: principal component analysis (PCA). We aim to understand the downstream implications of this algorithm, centering on identifying undesirable, systematic issues that may emerge with respect to individuals who are impacted by the algorithm's decisions. We use the term "unfairness" to refer to such an issue, an issue that induces a negative or undesirable impact on an individual or a group of individuals.

PCA is a foundational technique for dimensionality reduction which has been widely employed in many domains [13, 23]. PCA extracts key features from datasets by projecting them onto *principal components*, which reduces the dimension while preserving critical information. PCA has many downstream applications, and what type of "unfairness issues" exist will heavily depend on the exact application. Therefore, we focus our work to one common application of recommendation systems.

Recommendation systems and collaborative filtering. We use the running example of the LastFM music platform, where users listen to music by various artists (we refer to artists as *items*). In this context, the goal of a recommendation system is to help users discover artists that they would enjoy listening to. Collaborative filtering (CF) is a popular approach for recommendations that relies on using data on user-item preferences for a large number of users and finding patterns within these preferences. Dimensionality reduction methods, and PCA in particular, is a commonly used technique for CF (e.g., [10, 17, 19]). This paper focuses on the impact of using PCA for CF, and specifically, we focus on identifying unfairness issues at the *item*-level.

Contributions. We identify mechanisms of the PCA algorithm that can induce a negative effect for items in the context of recommendations, and then we develop an approach that tackles these unfairness mechanisms. We summarize our main contributions.

(1) We identify two mechanisms in which PCA may introduce an undesirable item-level impact within the context of CF.
   (a) The first mechanism is that the leading components of PCA often lack meaningful information related to less popular items. This may lead to fewer or worse-quality recommendations with respect to these less popular items.
   (b) The second mechanism is that in the existence of highly popular items, the leading components of PCA can each contain information about a *single* popular item, rather than capturing similarities between items. Such components are not useful for the sake of CF, as they do not contain any "collaborative" information; this can adversely impact the recommendations related to the highly popular items.

   We demonstrate both of these mechanisms empirically through the LastFM dataset, summarized in Figure 1, as well as theoretically on a stylized class of matrices.
(2) We propose a computationally efficient algorithm called *Item-Weighted PCA*, which optimizes an objective function that uses item-specific weights. The weights are given as input to the algorithm, hence this algorithm provides a framework for optimizing with any set of item-specific weights.
   (a) In a stylized class of matrices where popular and unpopular items are separated in a block-diagonal fashion, we show that *Item-Weighted PCA* with a specific set of weights minimizes the sum of normalized reconstruction errors across the two blocks.
   (b) We consider two natural benchmark algorithms, vanilla PCA, as well as column-normalized PCA, which normalizes each column of the matrix before performing PCA. In the stylized class of matrices, we show that both of these benchmark algorithms are a special case of *Item-Weighted PCA* with a specific set of weights. We then show that

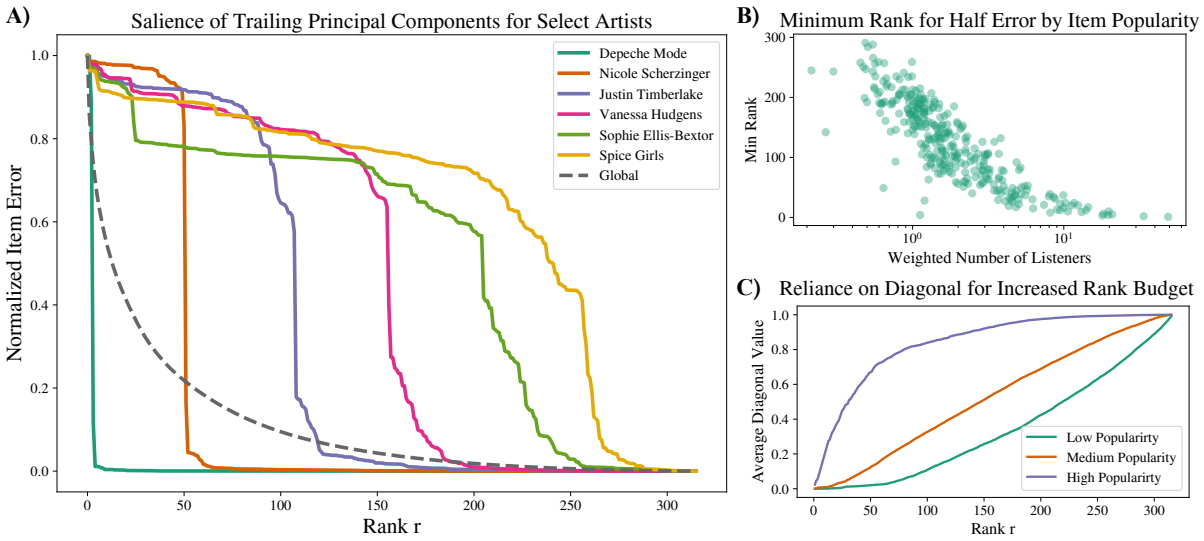


Figure 1: These figures are generated from computing vanilla PCA on the LastFM dataset for varying values of the rank r. Subfigure (A) shows the normalized item error as a function of the rank for six different artists, as well as the overall error in the dotted line. Subfigure (B) shows the relationship between an artist's popularity (weighted number of listeners) and the number of principal components needed to half the initial item reconstruction error. Subfigure (C) shows the average diagonal value of the projection matrix outputted by PCA, where artists are grouped by their popularity.

the approach of setting the weights to be inversely proportional to an item's norm is a *interpolation* between the two benchmark algorithms. We use this weighting procedure for all of our numerical experiments.

(3) We present empirical results demonstrating that our algorithm yields improved collaborative filtering recommendations compared to PCA baselines. Interestingly, we characterize how our algorithm improves recommendation quality for both popular and less popular artists.

We conclude with a discussion of limitations and recommended use cases for our algorithm.

1.1 Relation to Fair PCA Literature

While we provide a more extensive literature review in Section 5, we believe it is important to describe the connection of our work to the existing literature that studies fairness in the context of PCA.

Brief summary of literature. The existing literature on fair PCA can be summarized as imposing a fairness constraint on the PCA problem and developing a new algorithm to satisfy this constraint. Specifically, existing works assume that the set of users is partitioned into pre-defined groups (e.g., race, gender). There are a series of papers [15, 24, 26, 27] that define fairness as enforcing that the reconstruction error across groups of users to be "balanced", for different definitions of balance. Alternatively, [21] defines the output of a PCA algorithm as fair if the group label cannot be inferred from the projected data, while [18] aims to minimize the difference in the conditional distributions of the projected data.

Comparison to our work. Table 1 summarizes the differences between our work and existing literature. One difference is that prior works focus on *user*-level fairness with pre-defined groups, whereas we focus on *item*-level fairness, with no reliance on group labels.

However, there is also a major difference in the *motivation* of our work compared to existing works that induce a distinction in the types of situations that the works apply to. Specifically, the methods from existing works address situations where an algorithm designer knows, a priori, that they would like to enforce a certain type of fairness constraint. That is, there is an *external constraint* that deems a particular fairness notion necessary, and these fairness constraints are *generic*, in the sense that they can be defined in a general machine learning context.

On the other hand, the motivation of our work is to *identify* unfairness issues that arise specifically from the PCA algorithm. The issues that we identify are not generic machine learning issues, and hence they would not necessarily be issues that one would be concerned about a priori. Our work helps elucidate the black-box nature of the PCA algorithm and contributes to situations where one does not have a particular fairness notion in mind but would like to understand what types of issues can arise from this specific algorithm.

Analogs of this distinction appear in other areas. For example, in prediction, the seminal work of [11] studies how to learn a classifier with an external fairness constraint (equality of opportunity). In contrast, [4, 16] also study fairness in prediction, but the goal is to identify the reasons why bias may arise in a prediction setting, rather than developing algorithms that satisfy a fairness notion.

Algorithm	User	Item	Labels	Fairness Notion
Olfat and Aswani [21], Lee et al. [18]	✓		✓	obfuscate group identifiability
Samadi et al. [26], Tantipongpipat et al. [27], Kamani et al. [15], Pelegrina and Duarte [24]	✓		✓	balance reconstruction error across groups
Item-Weighted PCA		✓		improve collaborative-filtering recommendations

Table 1: Comparison with existing papers studying fair PCA.

1.2 Background on PCA

Let $X \in \mathbb{R}^{n\times d}$ be a matrix of preferences over n users and d items. PCA applied to X projects the matrix into a r-dimensional space yielding an approximation matrix $\widehat{X}$, where $r \ll d$ is a user-determined rank hyperparameter. Formally, PCA solves:

$$\begin{aligned} &\underset{P=UU^T}{\operatorname{argmin}} \|X - XP\|_F^2 \\ &\quad \text{s.t.} \quad U \in \mathbb{R}^{d\times r}, U^T U = I_r \end{aligned} \tag{1}$$

The optimization is over projection matrices $P = UU^T$ where the columns of $U \in \mathbb{R}^{d\times r}$ form an orthonormal basis. The optimal projection matrix P^* minimizes the reconstruction error $\|X - \widehat{X}\|_F^2$ between the original matrix and the approximation, $\hat{X} = XP^*$.

Note that the approximation matrix $\hat{X} = XP^*$ is equivalent to taking the r-truncated Singular Value Decomposition (SVD) of X. Henceforth, when referring to collaborative filtering we refer to the problem of identifying a suitable projection matrix $P = UU^T$ where we refer to solution to Equation (1) as the *vanilla PCA* baseline.

2 UNFAIRNESS OF PCA FOR COLLABORATIVE FILTERING

In this section, we begin with a motivating empirical example illustrating two mechanisms in which PCA exhibits unfairness towards items for collaborative filtering. Then, we show that these mechanisms provably occur in a stylized class of matrices that represent user-item preferences.

2.1 Empirical Example: LastFM

Our motivating empirical example uses the lastfm-2k dataset [2] which records the number of times a user of the LastFM[1] music platform listened to their favorite artists. Specifically, if artist j is one of user i's top 25 artists, then X_{ij} is the number of times user i listened to artist j. Otherwise $X_{ij} = 0$. We use a dataset with $n = 920$ users and $d = 316$ artists. To account for heterogeneity in user listening volume we row-normalize the matrix. See the Experiments section for a detailed description of this dataset. We compute PCA on this matrix X for all possible values of the rank r, from 0 to d. Let $P_r \in \mathbb{R}^{d\times d}$ be the projection matrix corresponding to the output of PCA for rank r.

We now describe two ways in which PCA induces unfairness for the items (artists).

2.1.1 Mechanism 1: Unfairness for unpopular items. The overall reconstruction error, $\|X - XP_r\|_F^2$ decreases as r increases in a diminishing returns fashion: see the dashed grey line in Figure 1, Subfigure A. In fact, it can be shown that reconstruction error decreases by exactly σ_r^2 at rank r compared to $r-1$, where $|\sigma_1| \geq \cdots \geq |\sigma_d|$ are the ordered singular values of X (see Theorem 8 in the Appendix).

However, this pattern of diminishing returns does not occur at the individual item level. We define the *normalized item error* for item j as $\|X_{\cdot j} - XP_{r,\cdot j}\|_2^2 / W_j$, where $X_{\cdot j}$ is the j'th column of X, $P_{r,\cdot j}$ is the j'th column of P_r, and $W_j = \|X_{\cdot j}\|_2^2$ is a normalizing factor. Subfigure A in Figure 1 plots the normalized item error for six individual artists (items), which displays the large heterogeneity in how the errors decrease as a function of the rank. For each artist, the normalized error is initially 1 when the rank is 0 since $P = 0$, and drops sharply after some threshold rank is reached, where this threshold varies greatly by the artist. Certain artist such as Jessica Simpson requires the rank to be over 200 before their normalized error decreases below 80%.

In general, the leading components of PCA capture the artists who are popular. Subfigure B in Figure 1 shows the relationship between artist popularity, where the popularity for artist j is $\sum_{i=1}^n X_{ij}$ following row normalization, and the number of principal components needed to half the initial reconstruction error of $\|X_{\cdot j}\|_2^2$. The Subfigure shows that leading principal components greatly reduce reconstruction error for popular artists. The top-20% most popular artists require 36 components, on average, to half their error while the bottom 80% requires 147 of 316 components.

2.1.2 Mechanism 2: Unfairness for popular items. We now describe a completely different mechanism that negatively impacts popular items. The previous mechanism showed that the leading components favor the popular items. However, we find that the leading components can become *specialized* in *individual* items, which has undesirable consequences in the context of collaborative filtering.

Recall that PCA outputs a projection matrix $P \in \mathbb{R}^{d\times d}$. We claim that it is undesirable for item j for the diagonal entry, P_{jj}, to be close to 1 at low values of r, which is the case for popular artists as seen in Subfigure C of Figure 1.

For an artist j, the approximation of its listening count for user i is $\hat{X}_{ij} = \sum_{k=1}^d X_{ik} P_{kj}$. Then, for an item $k \neq j$, the entry P_{kj} can be interpreted as a "similarity" between items j and k. A non-zero entry for P_{kj} implies that the preference towards artist k contributes to the reconstructed preference towards item j.

Now, if it is the case that the diagonal entry is 1 ($P_{jj} = 1$) and $P_{kj} = 0$ for all $k \neq j$, we recover a perfect reconstruction ($\hat{X}_{ij} = X_{ij}$). However, this implies that the reconstructed preference of item j is simply the original preference towards item j, which is not useful information in the context of collaborative filtering. This does not give us a way to infer whether a user will like item j given their preferences over other items. The diagonal entry P_{jj} being close

[1] http://www.lastfm.com

to 1 implies that most of the reconstructed value for $\hat{X}_{ij}$ is coming from X_{ij}.

2.2 Theoretical Result

We demonstrate that PCA exhibits the above two phenomena in a class of matrices that represent user-item preferences, where a subset of items is highly popular. We consider a sequence of systems of increasing size, where both the number of users and items is growing. Concretely, consider a sequence of matrices $\{X_n\}_{n\geq 1}$, where $X_n \in \{0,1\}^{n\times d_n}$ and $d_n = o(n)$. The (i,j)'th entry of X_n is 1 if user i likes item j, and 0 otherwise.

We assume that the items can be partitioned into two classes: popular items and unpopular items. We assume that the first M_n items are the popular items for X_n, for $M_n \leq d_n$, that satisfy the following assumption.

Assumption A (Popular items). *Let $X'_n \in \{0,1\}^{n\times d_n}$ be a copy of X_n where all entries in columns $j > M_n$ are set to zero. Then, we assume that the M_n'th largest singular value of X'_n, which we denote by $s_{M_n}(X'_n)$, grows as $\Omega(\sqrt{n})$.*

Note that Assumption A is satisfied with high probability if all entries of X'_n are i.i.d. mean zero subgaussian random variables with unit variance; see Theorem 1.1 in Rudelson and Vershynin [25] and Figure 6 in the Appendix for empirical validation.

Next, we assume that for unpopular items, the number of users that like the item is a constant.

Assumption B (Unpopular items). *There exists a constant K such that for all n, $\sum_{i=1}^{n}(X_n)_{i,j} \leq K$ for any $j > M_n$.*

Then, we show that PCA on the matrix X_n using the top M_n principal components admits the two undesirable mechanisms. Let $I_{n,M_n} \in \mathbb{R}^{d(n)\times d(n)}$ be the matrix where all entries are zero except for the first M_n diagonal entries, which are 1.

Theorem 1. *Let $P_n \in \mathbb{R}^{d_n\times d_n}$ be the projection matrix given by performing PCA on matrix X_n, taking the largest M_n principal components. Then, $||P_n - I_{n,M_n}||_F \to 0$ as $n \to \infty$.*

Theorem 1 states that as the system gets large, the projection matrix outputted by PCA with M_n components converges to the I_{n,M_n} matrix. The projection matrix being $P = I_{n,M_n}$ demonstrates both undesirable mechanisms. The proof makes use of the Davis-Kahan theorem from perturbation theory, which can be found in the Appendix.

Firstly, all columns $j > M_n$ that represent the less popular items are the 0 vector in the projection matrix; i.e. the projection does not contain *any* information about item j. Then, the reconstruction, $\hat{X}_{\cdot j}$ will also be the 0 vector; that is, the reconstructed preference of every user to every unpopular item is outputted to be 0.

Next, fix a popular item $j \leq M_n$. Then, column j of the projection matrix approaches e_j, the unit vector with 1 in the j'th entry. Then, the reconstruction of the preference of user i for item j, $\hat{X}_{ij}$, is exactly X_{ij}. That is, the reconstruction for the (i,j)'th entry just "reads" the value that was there in the original matrix. This provides a perfect reconstruction, but this provides no useful information in the context of collaborative filtering. The reconstruction only provides non-zero values to entries that already existed in the original matrix, which does not serve the purpose of using this method as a recommendation tool. A projection matrix that is useful for recommendations should contain many non-zero entries for column j: then, the preference of user i towards item j can be inferred through the existing preferences of user i towards *other* items $k \neq j$.

3 ITEM-WEIGHTED PCA

We propose an algorithm named *Item-Weighted PCA* that counters the unfairness mechanisms introduced in the previous section. We will formally state the problem we aim to solve and present *Item-Weighted PCA* as an algorithm solving the problem. Then, on a stylized class of matrices, we provide a theoretical justification for this approach, and we also show that two baseline approaches are a special case of *Item-Weighted PCA*.

3.1 Algorithm Description

3.1.1 Problem Statement. Let $X \in \mathbb{R}^{n\times d}$ be an input matrix, where entries can be positive or negative and missing values are set to zero, $r \leq \min\{n,d\}$ be a rank parameter, and $S \in \{-1,0,+1\}^{n\times d}$ be the *sign matrix* of X, where $S_{ij} = 1$ for positive X_{ij}, -1 for negative X_{ij}, and 0 when $X_{ij} = 0$. Let $w_j \geq 0$ for $j \in [d]$ be item-specific weights. We aim to solve the following problem:

$$\underset{P=UU^\top}{\operatorname{argmax}} \sum_{j=1}^{d} w_j \ \langle S_{\cdot j}, \hat{X}_{\cdot j}\rangle \tag{2}$$

$$\text{s.t.} \quad U \in \mathbb{R}^{d\times r},\ U^T U = I \tag{3}$$

where $\hat{X}_{ij} = \left\langle X_{i\cdot}, P_{\cdot j}\right\rangle \forall i, j$.

Note that the weights w_j must be given as input. In all of our experiments, we use the weights $w_j = 1/\|S_{\cdot j}\|_2$ In Section 3.2, we study a simple class of matrices where we specify how the weights should be chosen.

3.1.2 Algorithm. We propose the algorithm *Item-Weighted PCA*, which solves (2)-(3) by relaxing the feasible set. Instead of constraining to projection matrices $P = UU^T$, *Item-Weighted PCA* relaxes to optimize over positive semi-definite matrices (PSD) with bounded trace and eigenvalues and solves for an extreme-point optimal solution to the following Semi-Definite Program (SDP):

$$\underset{P}{\operatorname{argmax}} \sum_{j=1}^{d} w_j \ \langle S_{\cdot j}, \hat{X}_{\cdot j}\rangle \tag{4}$$

$$\text{s.t.} \quad \operatorname{tr}(P) \leq r, 0 \preceq P \preceq 1 \tag{5}$$

We observe that the set of PSD matrices with trace $\leq r$ and eigenvalues $\in [0,1]$ is a superset of rank r projection matrices. In the Appendix, we prove that the extreme-point optimal solution *Item-Weighted PCA* yields is indeed a projection matrix and thus solves the original problem.

Theorem 2. *Item-Weighted PCA is a polynomial-time algorithm to solve the optimization problem of* (2)-(3).

3.1.3 Discussion. We now describe the intuition and motivation of this algorithm, and in Section 3.2, we provide a theoretical justification for a special class of matrices.

Recall that vanilla PCA finds the projection matrix of rank r that minimizes the overall reconstruction error, $||X - XP||_F^2$. Then, *Item-Weighted PCA* makes the following modifications to vanilla PCA:

(a) We use the *sign matrix* S of X, which discards the magnitude of the original entries.
(b) The objective function uses *item-specific weights*, w_j.
(c) Instead of minimizing reconstruction error between the columns $S_{\cdot j}$ and $\hat{X}_{\cdot j}$, we maximize the inner products between the two vectors.

Modification (a). The motivation for (a) is a normalization of the original matrix that aligns with the downstream goal of *recommendations*, rather than *reconstruction*. That is, the goal of recommendations is to identify the (i, j) pairs where user i would enjoy item j, rather than reconstructing the exact entries X_{ij}. Because the entry magnitudes often vary greatly across users and contain outliers, using the sign matrix effectively introduces a normalization across all entries.

Modification (b). The item-specific weights aim to address both of the unfairness mechanisms. Suppose we use the weights $w_j = 1/\|S_{\cdot j}\|_2$, which we use for all of our experiments. Since less popular items have a smaller norm, this normalization up-weights these items, directly addressing the issue of unfairness towards less popular items (Mechanism 1). Moreover, this normalization also *down-weights* the significance of the highly popular items in the objective, which also addresses Mechanism 2. Recall that Mechanism 2 occurs when one of the components of PCA *specializes* in representing a *single* item. Since all items are effectively treated equally in (2), if the number of components is small (i.e. rank is small), then one cannot "afford" to dedicate one component to a single item – it is more efficient if each component contained information about multiple items.

Modification (c). Given modification (b), a natural alternative objective would be to keep the same error metric as vanilla PCA (square of entry-wise differences), with column-specific weights w_j, minimizing the least squares objective $\sum_{j=1}^{d} w_j \|S_{\cdot j} - \hat{X}_{\cdot j}\|_2^2$. Unfortunately, this objective does not yield a computationally efficient method. The allure of the objective (2) is that it is *linear* in P, which is not the case in the least squares objective; hence Theorem 2 would not hold. Therefore, the motivation for (c) is strictly for computational efficiency.

One interpretation of the $w_j \langle S_{\cdot j}, \hat{X}_{\cdot j} \rangle$ term in the objective (2) is an approximation to the *cosine similarity* between columns $S_{\cdot j}$ and $\hat{X}_{\cdot j}$. The exact cosine similarity would include an additional $1/\|\hat{X}_{\cdot j}\|_2$ term, hence using the exact cosine similarity would incorporate non-linearities into the objective, which would again be undesirable.

Note that because of modification (c), the objective (2) does not at all aim to reconstruct the original matrix X. However, it is possible to add constraints to enforce a small error if desired. Suppose $E_r = ||\hat{X}^{\text{PCA}} - X||_F^2$ is the reconstruction error of the vanilla PCA solution (which is the smallest possible reconstruction error). Then, one can add a constraint to the optimization (2)-(3) of the form $||\hat{X} - X||_F^2 \leq (1 + \alpha)E_r$ for some parameter $\alpha > 0$, so that the reconstruction error of the output $\hat{X}$ is at most a $(1 + \alpha)$ factor of E_r. In the Appendix, we show that Theorem 2 holds with the added constraint.

3.2 Theoretical Result and Comparison with Baseline Algorithms

We show that for a stylized class of matrices, *Item-Weighted PCA* yields the optimal solution to a popularity-normalized loss function. For the same class of matrices, we show that two baseline PCA algorithms are instantiations of *Item-Weighted PCA* with a specific set of weights. We then instantiate *Item-Weighted PCA* with weights that interpolate between the two baselines. In a specific setting, such an instantiation of *Item-Weighted PCA* balances popular and unpopular items, while the baselines offer two extremal solutions. The proofs for all propositions and theorems are included in Appendix A.3.

3.2.1 Optimality of Item-Weighted PCA. As in Theorem 1, we consider binary preference matrices $X \in \{0, 1\}^{n \times d}$ in which there are d_p popular items, corresponding to columns $I_p = \{1, \ldots, d_p\}$ and d_u unpopular items, corresponding to columns $I_u = \{d_p + 1, \ldots, d\}$.

We make the following assumption on X:

Assumption C (Exclusivity). *Each user likes either only popular items or only unpopular items.*

Let $\mathcal{B}$ be the set of binary matrices that satisfy Assumption C. By constraining users to like only one class of items, we ensure that individual principal components correspond exclusively to either popular or unpopular items.

In light of the imbalance in item popularities, given a matrix X, we introduce the following objective function that normalizes item reconstruction error by group popularity, quantified as the number of ratings for all items in the group:

$$l(P) = \left(\frac{\|X_p - \widehat{X}_p\|_F}{\|X_p\|_F} \right)^2 + \left(\frac{\|X_u - \widehat{X}_u\|_F}{\|X_u\|_F} \right)^2 \quad \text{where} \quad \widehat{X} = XP \tag{6}$$

In Equation (6), X_p denotes a copy of X with entries for all unpopular items set to 0 and X_u denotes a copy of X with ratings for popular items set to zero.

Theorem 3 (*Item-Weighted PCA* Optimality). *For $X \in \mathcal{B}$, Item-Weighted PCA yields the optimal solution for the popularity-adjusted loss function in Equation* (6) *when $w_j = \|X_p\|^{-2} \forall j \in I_p$ and $w_j = \|X_u\|^{-2} \forall j \in I_u$.*

Henceforth, we will call the weights in Proposition 3 the *proper* weights w_j.

Remark 1. *For $X \in \mathcal{B}$, there is a closed-form solution to minimize Equation 6 but for general X there is not a closed-form solution.*

3.2.2 Baseline Algorithms as a Special Case. We compare against two baselines: vanilla PCA and column-normalized PCA which scales each column of X to be unit norm before performing vanilla PCA. In the case of matrices in $\mathcal{B}$, we can interpret vanilla PCA and column-normalized PCA as specific instantiations of *Item-Weighted PCA* given:

Assumption D (Constant Popularity). *For all popular items, there are n_p users that like the item, and for all unpopular items, there are n_u users that like the item, where $n_u < n_p$.*

Proposition 4. *For $X \in \mathcal{B}$ and satisfying Assumption D, vanilla PCA instantiates Item-Weighted PCA with $w_j = 1\ \forall j \in I_p$ and column-normalized PCA instantiates Item-Weighted PCA with $w_j = n_p^{-1}\ \forall j \in \{1, \ldots, d_p\}$ and $w_j = n_u^{-1}\ \forall j \in I_u$.*

Placing the instantiations in the context of the proper weights identified in Theorem 3, we see that vanilla PCA yields the proper weights when $d_u = \frac{n_p}{n_u} d_p$. Column-normalized PCA is optimal when $d_p = d_u$. As the baselines use weights that are not a function of d_p, d_u, which are generally unknown, the baselines are suboptimal in minimizing Equation (6) in all other settings.

To show that *Item-Weighted PCA* provides a flexible framework, we define an instantiation that interpolates between vanilla PCA and column-normalized PCA. Let *Interpolate-Item-Weighted PCA* be the instantiation in which $w_j = \sqrt{n_p}\ \forall j \in I_p$ and $w_j = \sqrt{n_u}\ \forall j \in I_u$

We now provide a concrete instance in which *Interpolate-Item-Weighted PCA* balances popular and unpopular items while the baselines yield extreme, undesirable outcomes. For the specific example, we introduce an additional assumption:

Assumption E (Exponential Decay). *$X_p^T X_p$ and $X_u^T X_u$ are both of rank r and their respective eigenvalues decay exponentially such that for each matrix, the i^{th} largest eigenvalue $\lambda_i = \beta^{-(i-1)}\lambda_1$, where $\beta > 1$ and $i \leq r$.*

Theorem 5. *For any binary preference matrix $X \in \mathcal{B}$ satisfying Assumption E, if $\frac{n_u}{n_p} < \beta^{-2(r-1)}$ and $d_u = \sqrt{\frac{n_p}{n_u}} d_p$, then the leading r vanilla PCA components are V_p; the leading r column-normalized PCA components are V_u. For Interpolate-Item-Weighted PCA, half of the leading components are in V_p and the other half is in V_u.*

Theorem 5 states that when the popularity gap is large enough and there are sufficiently many unpopular items, for a rank r projection, vanilla PCA only reconstructs popular items whereas column-normalized PCA only reconstructs unpopular items. *Interpolate-Item-Weighted PCA*, on the other hand, reconstructs both popular and unpopular items in parallel. We observe that the above conditions mimic real-world settings in which there is a long tail of unpopular items.

4 EXPERIMENTS

4.1 Datasets

LastFM. We use the lastfm-2k dataset of user listening counts introduced in our motivating example where entry ij is the number of times user i listened to artist j if artist j is one of user i's top-25 most-listened artists, otherwise $X_{ij} = 0$. We filter the dataset to keep only artists with at least 50 top listeners and then users with at least 20 listening counts among the remaining artists, leaving a 920×316 matrix. We row normalize the listening counts for all users.

MovieLens. We use the MovieLens-1M dataset in which users provide ratings for movies on a scale from $1-5$ [12]. To reduce the number of movies while preserving the heterogeneity in interests, we filter for the top 30 movies among all 17 genres, omitting duplicates. We also filter for the top 2000 users in terms of the number of ratings provided yielding a 2000×308 data matrix. To capture the valence of the ratings, we re-map the original ratings of $1-5$ to $\{-2, -1, 1, 2, 3\}$, respectively.

These two datasets cover both explicit feedback in the form of user-provided ratings in the Movielens dataset as well as implicit feedback in the form of listening counts in the LastFM dataset.

4.2 Evaluation Methodology

The goal of our evaluation is to assess whether our *Item-Weighted PCA* improves recommendations from the item perspective. As such, we introduce the following methodology: given a data matrix $X \in \mathbb{R}^{n \times d}$ in which missing values are set to zero, we execute *Item-Weighted PCA*, yielding a $d \times d$ projection matrix P of rank r, which is a pre-determined integer rank budget. To test, for each item, we use P to classify all users as either "relevant" or "irrelevant" to the item. To avoid accessing the true user ratings, let us define P' as P with the diagonal entries zeroed out. Then, for an item j, the user scores are $(XP')_j$. Zeroing out the diagonal forces the recommendation to be based on the user's ratings for items that are similar to item j other than item j itself. For a given rank budget r, the evaluation score is the average AUC over all items:

$$\frac{1}{d} \sum_{i-1}^{d} \text{AUC}\left(XP'_j, y_j\right) \qquad \text{(Item AUC-ROC)}$$

In the above evaluation metric, y_j are the true binary labels and for both datasets, we define $y_j = X_j > 0$. For Movielens this is defined on the pre-processed ratings so a positive value corresponds with a raw rating ≥ 3. We compare our method against vanilla PCA and column-normalized PCA.[2]

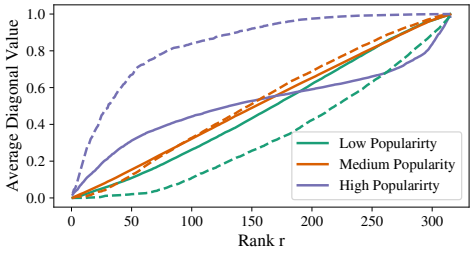
4.3 Results

We present four results from our evaluation: 1) *Item-Weighted PCA* mitigates leading components specializing in individual items; 2) At the item level, *Item-Weighted PCA* improves the classification of relevant users from less relevant users following projection; 3) The performance gains are observed at every item popularity level; 4) Compared to vanilla PCA, *Item-Weighted PCA* is more robust to uniformly randomly missing data.

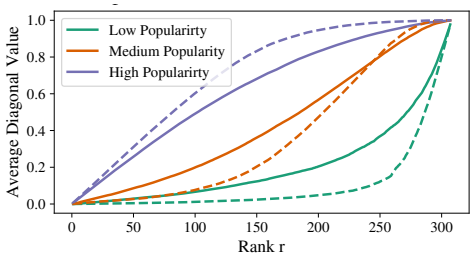
4.3.1 Reduced Specialization. We previously illustrated a vanilla PCA unfairness mechanism in which leading principal components specialize in individual popular artists as evidenced by large diagonal entries in P_r for low values of r. In Figure 2, we show that the average diagonal entries vary less by popularity in *Item-Weighted PCA* (solid) than in vanilla PCA (dashed). The difference is most noticeable in the reduction of diagonal entries corresponding to high-popularity artists in LastFM.

4.3.2 Overall Performance. In aggregate across all users in the LastFM and Movielens datasets, *Item-Weighted PCA* improves item-level classification performance compared to vanilla PCA at all rank budget values, as shown in Figure 3. In the figure, the y-axis is the average value of our Item AUC-ROC metric across all items. The curves for all algorithms decrease for large values of r because our evaluation metric zeros out the diagonal, and for large values of r collaborative filtering is not needed as P approaches the identity matrix. The performance improvement is most noticeable for LastFM, where *Item-Weighted PCA* also dominates the column

[2]We ran all of our experiments in Python on a machine with Intel Xeon E5-2690 CPUs, 2.60 GHz, 30 MB of cache.

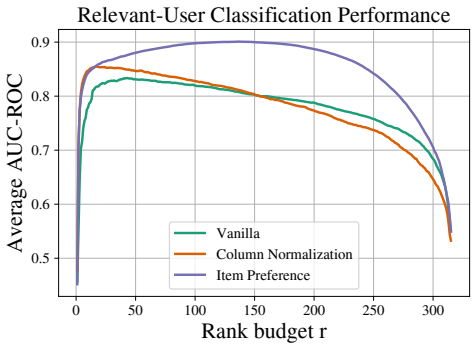


(a) LastFM

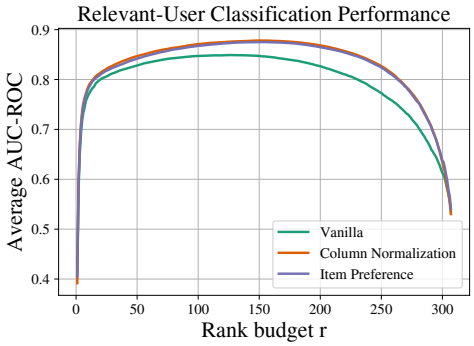


(b) MovieLens

Figure 2: *Item-Weighted PCA* (solid) reduces the unfairness mechanism identified in vanilla PCA (dashed) in which leading components specialize in individual items. High diagonal entries suggest specialization.



(a) LastFM



(b) MovieLens

Figure 3: For both LastFM and Movielens, *Item-Weighted PCA*, improves the ability for collaborative filtering to identify relevant listeners for each artist.

normalization baseline, especially for $r \in [50, 250]$. For Movielens, normalizing the columns performs comparably with our algorithm.

4.3.3 Performance by Item Popularity. *Item-Weighted PCA* is able to increase user classification performance for all item popularity groups instead of increasing performance for one group at the expense of another. Figure 4 shows that the user-classification performance increased for all popularity groups relative to vanilla PCA. The popularity groups were defined to approximately be of equal size.

The collective benefit illustrates the limitations of vanilla PCA. The recommendation quality for high-popularity items is lowest in vanilla PCA for both datasets because these items rely heavily on the diagonal values of the projection matrix and rely less on item similarities. By limiting the focus on any individual item, the overall item-level similarities captured in P are improved which benefits items of all popularity levels.

4.3.4 Robustness to Missing Data. We also assess our algorithm's robustness to missing data. In Figure 5, we plot the recommendation performance as training data points are gradually set to 0, where α is the fraction of training data points that have been uniformly randomly removed, for a fixed value of $r = 106$. In the case of LastFM, our algorithm outperforms both baselines for $\alpha < 0.6$. Whereas for MovieLens, our algorithm is not as robust as column normalization, though all three algorithms perform similarly for all values of α. We chose to fix $r = 106$ because our algorithm outperforms the baselines in Figure 3 when all data are available. In the Appendix, we include robustness results for all values of r.

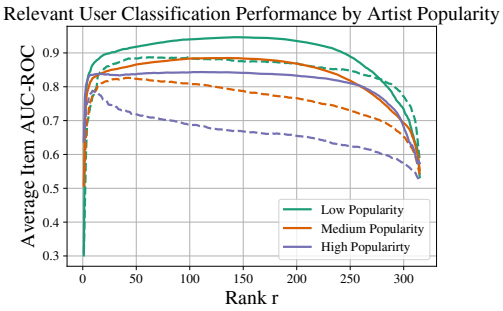
5 RELATED WORK

5.1 Fair PCA

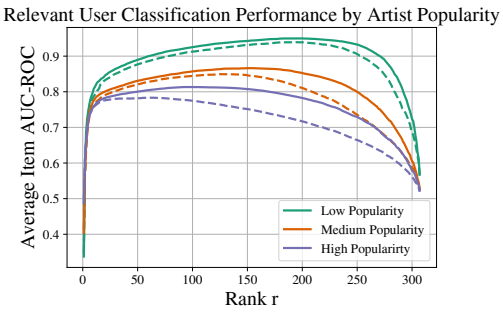
We provide further background on past fair PCA works that balance the reconstruction errors across groups of users [15, 24, 26, 27]. Many existing approaches solve a convex optimization problem of the following structure: let X_g denote the sub-matrix of X comprising of all individuals in group $g \in \{1, 2, \cdots, G\}$, f_P be the reconstruction error using projection matrix P, A be an aggregation function, and U be an $n \times r$ matrix with orthonormal columns; then, existing fair PCA algorithms can be generalized as:

$$\underset{P=UU^T}{\text{argmin}} \quad A\left(f_P\left(X_1\right), f_P\left(X_2\right), \cdots, f_P\left(X_G\right)\right) \tag{7}$$

By considering the reconstruction error of individual groups, existing fair PCA algorithms can ensure more balanced approximation quality. Common instances of the aggregation function A are the max function or the product function $\prod_{g=1}^{G} f_P\left(X_g\right)$. A cost of the above convex optimization approach is that the solution projection matrix is not guaranteed to be of rank r, where the rank increase is a function of the number of groups. More recent work has also presented non-convex optimization methods [18].

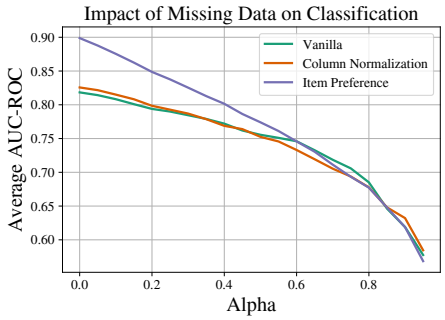


(a) LastFM

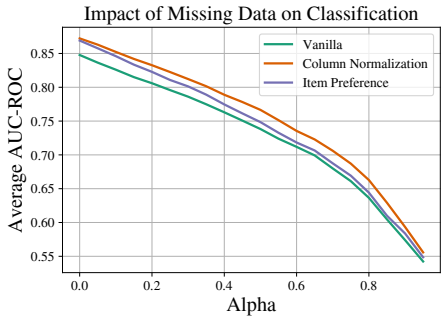


(b) MovieLens

Figure 4: *Item-Weighted PCA* (solid) is able to improve recommendation performance for items of all popularity levels relative to vanilla PCA (dashed). The improvement arises from projection matrices that better capture item similarities for collaborative filtering.



(a) LastFM



(b) MovieLens

Figure 5: The above charts show the robustness of the PCA algorithms as training examples are gradually removed at a fixed value of $r = 106$. α is the fraction of training examples that are removed (set to zero). *Item-Weighted PCA* is more robust than both baselines for LastFM and performs comparably with the baselines for MovieLens.

5.2 Trustworthy Recommender Systems

Fairness in the context of recommender systems and rankings has frequently been posed as a two-sided problem, balancing the interests of users and items/producers [6, 22]. On the user side, fairness definitions typically center on user utility either at the group level, defined by demographics [8], or at more granular levels, such as the notion of envy-freeness [7, 14], which states that no user should prefer another user's recommendations. In contrast, our work is more connected to notions of item fairness which are defined in terms of item exposure [3].

Additional prior work has focused on improving long-tail recommendations. Because many recommendation datasets feature a large number of items but a small number of highly popular "head" items, recommender systems are prone to popularity bias in disproportionately recommending popular items [20]. Over time this can lead to a "rich getting richer" effect, which is undesirable because many of the unpopular "tail" items may be desirable [9]. While many trustworthy recommender systems works are focused on introducing new exposure to unpopular items, our work is more focused on preserving existing preferences for less popular items.

6 LIMITATIONS

We discuss several known and potential limitations of our algorithm *Item-Weighted PCA*. First, the SDP has a runtime complexity of $O\left(d^{6.5}\right)$ [1], which means that *Item-Weighted PCA* can be prohibitively slow for large values of d. Second, it is possible that *Item-Weighted PCA* can overfit the input matrix in cases where the solution matrix is used to project out-of-sample matrices. Last, compared to vanilla PCA, the projection components are not ordered, so it is not possible to deduce P_r from P_{r+1}.

7 CONCLUSION

By analyzing within the context of collaborative filtering and recommender systems, we identify two mechanisms of unfairness in PCA. First, information relevant to less popular items is lacking in the leading components. Second, the leading components specialize in individual popular items instead of capturing similarities between items. These mechanisms arise from heterogeneity in item popularities and do not require external group labels to analyze. We illustrate the consequences of these mechanisms in a motivating real-world example and show that the mechanisms provably occur in a stylized setting. To mitigate unfairness, we introduce an algorithm, *Item-Weighted PCA*, that is designed to preserve user preferences for both popular and less popular items. *Item-Weighted PCA* is optimal in a stylized setting and our evaluations show that *Item-Weighted PCA* not only improves recommendations in aggregate but benefits both popular and less popular items.

REFERENCES

[1] Aharon Ben-Tal and Arkadi Nemirovski. 2001. *Lectures on Modern Convex Optimization*. SIAM.

[2] Iván Cantador, Peter Brusilovsky, and Tsvi Kuflik. 2011. 2nd Workshop on Information Heterogeneity and Fusion in Recommender Systems (HetRec 2011). In *RecSys'11*. ACM, New York, NY, USA.

[3] L. Elisa Celis, Damian Straszak, and Nisheeth K. Vishnoi. 2018. Ranking with Fairness Constraints. In *ICALP'18*. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, Dagstuhl, Germany, 28:1–28:15.

[4] Irene Chen, Fredrik D Johansson, and David Sontag. 2018. Why is my classifier discriminatory? *Advances in neural information processing systems* 31 (2018), 3543–3554.

[5] Chandler Davis and William Morton Kahan. 1970. The rotation of eigenvectors by a perturbation. III. *SIAM J. Numer. Anal.* 7, 1 (1970), 1–46.

[6] Virginie Do, Sam Corbett-Davies, Jamal Atif, and Nicolas Usunier. 2021. Two-sided fairness in rankings via Lorenz dominance. In *NeurIPS'21*. Curran Associates, Inc., 8596–8608.

[7] Virginie Do, Sam Corbett-Davies, Jamal Atif, and Nicolas Usunier. 2022. Online Certification of Preference-Based Fairness for Personalized Recommender Systems. In *AAAI'22*. 6532–6540.

[8] Michael D. Ekstrand, Mucun Tian, Ion Madrazo Azpiazu, Jennifer D. Ekstrand, Oghenemaro Anuyah, David McNeill, and Maria Soledad Pera. 2018. All The Cool Kids, How Do They Fit In?: Popularity and Demographic Biases in Recommender Evaluation and Effectiveness. In *FAT*'18*. PMLR, 172–186.

[9] Chongming Gao, Kexin Huang, Jiawei Chen, Yuan Zhang, Biao Li, Peng Jiang, Shiqi Wang, Zhong Zhang, and Xiangnan He. 2023. Alleviating Matthew Effect of Offline Reinforcement Learning in Interactive Recommendation. In *SIGIR'23*. ACM, New York, NY, USA, 238–248.

[10] Ken Goldberg, Theresa Roeder, Dhruv Gupta, and Chris Perkins. 2001. Eigentaste: A constant time collaborative filtering algorithm. *information retrieval* 4 (2001), 133–151.

[11] Moritz Hardt, Eric Price, and Nathan Srebro. 2016. Equality of Opportunity in Supervised Learning. In *NIPS'16* (Barcelona, Spain). Curran Associates Inc., Red Hook, NY, USA, 3323–3331.

[12] F. Maxwell Harper and Joseph A. Konstan. 2015. The MovieLens Datasets: History and Context. *ACM Trans. Interact. Intell. Syst.* 5, 4 (2015), 19 pages. https://doi.org/10.1145/2827872

[13] Harold Hotelling. 1933. Analysis of a complex of statistical variables into principal components. *Journal of educational psychology* 24, 6 (1933), 417.

[14] Christina Ilvento, Meena Jagadeesan, and Shuchi Chawla. 2020. Multi-Category Fairness in Sponsored Search Auctions. In *FAT*'20'*. ACM, New York, NY, USA, 348–358.

[15] Mohammad Mahdi Kamani, Farzin Haddadpour, Rana Forsati, and Mehrdad Mahdavi. 2022. Efficient Fair Principal Component Analysis. *Mach. Learn.* 111, 10 (oct 2022), 3671–3702.

[16] Fereshte Khani and Percy Liang. 2020. Feature noise induces loss discrepancy across groups. In *ICML'20*. JMLR.org, 5209–5219.

[17] Dohyun Kim and Bong-Jin Yum. 2005. Collaborative filtering based on iterative principal component analysis. *Expert Systems with Applications* 28, 4 (2005), 823–830.

[18] Junghyun Lee, Gwangsu Kim, Matt Olfat, Mark Hasegawa-Johnson, and Chang D. Yoo. 2022. Fast and Efficient MMD-Based Fair PCA via Optimization over Stiefel Manifold. In *AAAI'22*. 7363–7371.

[19] Mehrbakhsh Nilashi, Othman bin Ibrahim, Norafida Ithnin, and Nor Haniza Sarmin. 2015. A multi-criteria collaborative filtering recommender system for the tourism domain using Expectation Maximization (EM) and PCA–ANFIS. *Electronic Commerce Research and Applications* 14, 6 (2015), 542–562.

[20] Xichuan Niu, Bofang Li, Chenliang Li, Rong Xiao, Haochuan Sun, Hongbo Deng, and Zhenzhong Chen. 2020. A Dual Heterogeneous Graph Attention Network to Improve Long-Tail Performance for Shop Search in E-Commerce. In *KDD'20*. ACM, New York, NY, USA, 3405–3415.

[21] Matt Olfat and Anil Aswani. 2019. Convex Formulations for Fair Principal Component Analysis. In *AAAI'19/IAAI'19/EAAI'19*. AAAI Press, Article 82, 8 pages.

[22] Gourab K Patro, Arpita Biswas, Niloy Ganguly, Krishna P. Gummadi, and Abhijnan Chakraborty. 2020. FairRec: Two-Sided Fairness for Personalized Recommendations in Two-Sided Platforms. In *WWW'20*. ACM, New York, NY, USA, 1194–1204.

[23] Karl Pearson. 1901. LIII. On lines and planes of closest fit to systems of points in space. *The London, Edinburgh, and Dublin philosophical magazine and journal of science* 2, 11 (1901), 559–572.

[24] Guilherme Dean Pelegrina and Leonardo Tomazeli Duarte. 2022. A novel approach for Fair Principal Component Analysis based on eigendecomposition. https://doi.org/10.48550/ARXIV.2208.11362

[25] Mark Rudelson and Roman Vershynin. 2009. Smallest singular value of a random rectangular matrix. *Communications on Pure and Applied Mathematics: A Journal Issued by the Courant Institute of Mathematical Sciences* 62, 12 (2009), 1707–1739.

[26] Samira Samadi, Uthaipon Tantipongpipat, Jamie Morgenstern, Mohit Singh, and Santosh Vempala. 2018. The Price of Fair PCA: One Extra Dimension. In *NIPS'18*. Curran Associates, Inc.

[27] Uthaipon (Tao) Tantipongpipat, Samira Samadi, Mohit Singh, Jamie Morgenstern, and Santosh Vempala. 2019. Multi-Criteria Dimensionality Reduction with Applications to Fairness. In *NIPS'19*. Curran Associates Inc., Red Hook, NY, USA, 11 pages.

[28] Yi Yu, Tengyao Wang, and Richard J Samworth. 2015. A useful variant of the Davis–Kahan theorem for statisticians. *Biometrika* 102, 2 (2015), 315–323.

Appendix for "When Collaborative Filtering is not Collaborative: Unfairness of PCA for Recommendations"

Anonymous submission

A PROOFS

A.1 Proof of Theorem 1

Let $X'_n \in \{0,1\}^{n \times d_n}$ be a copy of X_n where all entries in columns $j > M_n$ are set to zero. We will show that the projection matrix corresponding to performing PCA on X_n is close to the projection matrix of PCA on X'_n.

Let $C_n = X_n^\top X_n$ and $C'_n = X_n'^\top X'_n$. Let $U_n, U'_n \in \mathbb{R}^{d_n \times M_n}$ be the matrix whose columns correspond to the M_n normalized eigenvectors corresponding to the M_n largest eigenvalues of C_n and C'_n respectively.

Claim 6. $U'_n U_n'^\top = I_{n,M_n}$

Proof of Claim 6. Let $Y_n \in \mathbb{R}^{M_n \times M_n}$ correspond to the top-left block of C'_n. Let $V_n \in \mathbb{R}^{M_n \times M_n}$ have columns that are the eigenvectors of Y_n, where V_n is orthonormal (since Y_n is symmetric). Therefore, $V_n V_n^\top = I_{M_n}$ is the identify matrix. Now, C'_n is simply Y_n in its top left block, and all other entries are 0. Therefore, if $v \in \mathbb{R}^{M_n}$ is an eigenvector of Y_n, then the vector v padded with zeros, $(v, 0, \ldots, 0) \in \mathbb{R}^d$ is an eigenvector of C'_n. Therefore, each column of U'_n is simply an eigenvector v of Y_n, padded with 0's to make it a length d vector. The other eigenvectors of C'_n that do not have this form are the ones whose corresponding eigenvalue is 0, since 0 is an eigenvalue of C'_n with multiplicity $d_n - M_n$. ■

Now, we use a variant of the Davis-Kahan theorem [5] from Yu et al. [28]. Using the notation in Theorem 2 in Yu et al. [28], we let $r = 1$ and $s = M_n$. Then, using the fact that $||\sin\Theta(U, U')||_F = \frac{1}{\sqrt{2}}||UU^\top - U'U'^\top||_F$,

$$||UU^\top - I_{n,M_n}||_F \leq \frac{2\sqrt{2}\,||C - C'||_F}{\lambda_{M_n}(C')}, \tag{8}$$

where $\lambda_{M_n}(C')$ is the M_n'th largest eigenvalue of C'. Since all of the less popular items satisfy Assumption B, every entry in $C - C'$ is upper bounded by K. Therefore, $||C - C'||_F \leq d_n K$. Next, since the eigenvalues of C' correspond to the square of the singular values of X'_n, and since $s_{M_n}(X'_n) = \Omega(\sqrt{n})$, we have that $\lambda_{M_n}(C') = \Omega(n)$. Therefore, $||P_n - I_{n,M_n}||_F = O(d_n K/n)$, which approaches 0 as $n \to \infty$ since $d = o(n)$ and K is a constant. □

A.2 Proof of Theorem 2

To prove the theorem, we must show that extreme-point optimal solutions to the convex relaxation in Equations (4)-(5) (*Item-Weighted PCA*()) are optimal solutions for the problem statement in Equations (2)-(3) (the "original problem").

The relaxation in *Item-Weighted PCA* is over the feasible set. Instead of optimizing over rank r projection matrices $P = UU^T$, *Item-Weighted PCA* optimizes over PSD matrices with bounded eigenvalues and trace. Observe that any optimal solution to the problem posed in the original problem is a feasible solution for *Item-Weighted PCA*.

Claim 7. *Any optimal solution P^* to the original problem is a feasible solution for Item-Weighted PCA.*

Proof. The optimal solution P^* is a projection matrix that can be factorized as $P^* = U^* U^{*T}$. This factorization is also the eigendecomposition of the matrix where the eigenvalues are 1 with multiplicity r and 0 with multiplicity $d - r$. Since the trace of a matrix is the sum of its eigenvalues, the trace constraint $\text{tr}(P^*) = r$ in *Item-Weighted PCA* is satisfied. Further since all eigenvalues are $\in [0, 1]$ the eigenvalue constraints $0 \preceq P^* \preceq I_d$ are also satisfied. □

Now, we can prove the theorem if we can show that an extreme-point optimal solution to *Item-Weighted PCA* satisfies two properties (i) can be expressed as UU^T where $U^T U = I_r$ and (ii) can be found in polynomial time.

To show (i) we utilize the definition of an extreme point. An extreme point of a convex set is a point that is not a linear combination of two other points in the convex set. For the convex set defined in (5), an extreme point must have eigenvalues of 0 and 1.

Suppose there is an extreme point $P' = \sum_{i=1}^d \lambda_i u_i u_i^T$ where there exists a single fractional $\lambda_{i'} \in (0, 1)$. Then it is possible to define P' as the linear combination (average) of the matrices $P' + \epsilon u_{i'} u_{i'}^T$ and $P' - \epsilon u_{i'} u_{i'}^T$ where $\epsilon \leq \lambda_{i'} \leq 1 - \epsilon$. Note that when there is one fractional eigenvalue, the trace constraint is not tight since r is an integer, thus $P' + \epsilon u_{i'} u_{i'}^T$ is a feasible matrix.

If there are two or more fractional eigenvalues the matrix also cannot be an extreme point. Let $\lambda_1, \lambda_2 \in (0, 1)$. Then we define P' as the average of two matrices: $P' + \epsilon_1 \lambda_1 u_1 u_1^T - \epsilon_2 \lambda_2 u_2 u_2^T$ and $P' - \epsilon_1 \lambda_1 u_1 u_1^T + \epsilon_2 \lambda_2 u_2 u_2^T$ where $\epsilon_1 \lambda_1 = \epsilon_2 \lambda_2$. Note that the perturbations does not affect the trace of the matrix so the perturbed matrices are feasible even if the trace constraint is tight for P'.

Now, since all eigenvalues of extreme points for *Item-Weighted PCA* are 0 or 1 and the trace is the sum of eigenvalues, the rank of an extreme point is at most r to satisfy the trace constraint. Thus, an extreme point of *Item-Weighted PCA* can be eigendecomposed as UU^T where $U^T U = I_r$.

To show (ii) we utilize Theorem 1.8 from Tantipongpipat et al. [27] which states that for SDPs with a linear objective function, m linear (in)equality constraints, and eigenvalue constraints in Equation 5 an extreme-point optimal solution can be found in polynomial time.

Last we discuss the addition of an optional linear reconstruction error constraint and show that *Item-Weighted PCA* yields a projection matrix of rank at most d. From Theorem 1.8 in Tantipongpipat et al. [27], we have that all extreme point optimal solutions have rank at most r and can be found in polynomial time.

To show that an extreme point optimal solution is a projection matrix we must again show that the eigenvalues are integer. We prove by contradiction: consider an extreme point optimal solution $P' = \sum_{i=1}^d \lambda_i u_i u_i^T$ where there exists a single fractional eigenvalue $\lambda_{i'}$. We can show that such a point cannot be optimal because setting $\lambda_{i'} = 1$ would be feasible and improve the objective. The objective function can be written as a linear combination of the eigenvalues of P': $\sum_{i=1}^d c_i \lambda_i$. $c_{i'}$ must be positive, otherwise $P' - \lambda_{i'} u_{i'} u_{i'}^T$ would have a higher objective value. Thus increasing $\lambda_{i'}$ increases the

objective. Setting to 1 also decreases the reconstruction error given that the increase of any eigenvalue decreases reconstruction error. And the perturbed matrix $P' + (1 - \lambda_{i'})\, u_{i'} u_{i'}^T$ is feasible for the integer trace constraint given that the trace of P' is at most $r + \lambda_{i'} - 1$.

If there are two fractional eigenvalues, the argument with ϵ_1 and ϵ_2 can again be used to show P' is not an extreme point, where ϵ_1 and ϵ_2 are defined to preserve reconstruction error. □

A.3 Proofs for *Item-Weighted PCA* Analysis

A.3.1 Theorem 3.

Proof. Let us re-write the normalized reconstruction error for X_p in terms of a trace:

$$\left(\frac{\|X_p - \widehat{X}_p\|_F}{\|X_p\|_F}\right)^2 = \left(\frac{\|X_p - X_p P\|_F}{\|X_p\|_F}\right)^2 \tag{9}$$

$$= \|\frac{X_p}{\|X_p\|_F} - \frac{X_p}{\|X_p\|_F} P\|_F^2 \tag{10}$$

$$= \|\frac{X_p}{\|X_p\|_F}\|^2 - \mathrm{Tr}\left(\frac{X_p^T X_p}{\|X_p\|_F^2} P\right) \tag{11}$$

Observe that minimizing the normalized reconstruction error for popular items is equivalent to maximizing $\mathrm{Tr}\left(\frac{X_p^T X_p}{\|X_p\|_F^2} P\right)$. Repeating the same process for unpopular items, Equation 6 becomes:

$$l(P) = -\mathrm{Tr}\left(\frac{X_p^T X_p}{\|X_p\|_F^2} P\right) - \mathrm{Tr}\left(\frac{X_u^T X_u}{\|X_u\|_F^2} P\right) + C \tag{12}$$

Where C is a constant.

Now, we show that instantiating *Item-Weighted PCA* with the weights stated in the Theorem is equivalent to minimizing Equation 12. After instantiation, the objective for *Item-Weighted PCA* is equivalent to maximizing:

$$\sum_{j=1}^{d} w_j \left\langle S_{\cdot j}, \widehat{X}_{\cdot j} \right\rangle = \sum_{j \in I_p} \frac{1}{\|X_p\|_F^2} \left\langle S_{\cdot j}, \widehat{X}_{\cdot j} \right\rangle + \sum_{j \in I_u} \frac{1}{\|X_u\|_F^2} \left\langle S_{\cdot j}, \widehat{X}_{\cdot j} \right\rangle \tag{13}$$

$$= \sum_{j \in I_p} \frac{1}{\|X_p\|_F^2} \left\langle X_{\cdot j}, \widehat{X}_{\cdot j} \right\rangle + \sum_{j \in I_u} \frac{1}{\|X_u\|_F^2} \left\langle X_{\cdot j}, \widehat{X}_{\cdot j} \right\rangle \tag{14}$$

$$= \frac{1}{\|X_p\|_F^2} \mathrm{Tr}\left(X_p^T X_p\right) + \frac{1}{\|X_u\|_F^2} \mathrm{Tr}\left(X_u^T X_u\right) \tag{15}$$

Where for binary matrices $S = X$.

We have now shown that minimizing Equation 6, reformulated as Equation 12, is equivalent to *Item-Weighted PCA* with the given weights, which is equivalent to maximizing Equation 15. □

A.3.2 Proposition 4.

Proof. Vanilla PCA maximizes $\mathrm{Tr}\left(X^T X P\right)$. We can re-write the trace as the dot product $\langle X, XP \rangle$. Since $X = S$ for binary matrices, vanilla PCA maximizes $\left\langle S, \widehat{X} \right\rangle$, which is exactly equal to *Item-Weighted PCA* instantiated with $w_j = 1 \forall j \in [d]$.

For column normalization, we will show that instantiating *Item-Weighted PCA* with the given weights is equivalent to column normalization.

$$\sum_{j \in I_p} \frac{1}{n_p} \left\langle S_{\cdot j}, \widehat{X}_{\cdot j} \right\rangle + \sum_{j \in I_u} \frac{1}{n_u} \left\langle S_{\cdot j}, \widehat{X}_{\cdot j} \right\rangle \tag{16}$$

$$= \sum_{j \in I_p} \frac{1}{n_p} \left\langle X_{\cdot j}, \widehat{X}_{\cdot j} \right\rangle + \sum_{j \in I_u} \frac{1}{n_u} \left\langle X_{\cdot j}, \widehat{X}_{\cdot j} \right\rangle \tag{17}$$

$$= \frac{1}{n_p} \sum_{j \in I_p} \left\langle X_{\cdot j}, (XP)_{\cdot j} \right\rangle + \frac{1}{n_u} \sum_{j \in I_u} \left\langle X_{\cdot j}, (XP)_{\cdot j} \right\rangle \tag{18}$$

Observe that $\sum_{j \in I_p} \left\langle X_{\cdot j}, (XP)_{\cdot j} \right\rangle$ is equal to $\mathrm{Tr}\left(X_p^T X_p P\right)$ since $X_p^{(ij)} = 0 \; \forall j \in I_u$. An analogous claim can be made for the sum over unpopular columns. Thus, we have:

$$\sum_{j \in I_p} \frac{1}{n_p} \left\langle S_{\cdot j}, \widehat{X}_{\cdot j} \right\rangle + \sum_{j \in I_u} \frac{1}{n_u} \left\langle S_{\cdot j}, \widehat{X}_{\cdot j} \right\rangle \tag{19}$$

$$= \frac{1}{n_p} \sum_{j \in I_p} \left\langle X_{\cdot j}, (XP)_{\cdot j} \right\rangle + \frac{1}{n_u} \sum_{j \in I_u} \left\langle X_{\cdot j}, (XP)_{\cdot j} \right\rangle \tag{20}$$

$$= \frac{1}{n_p} \mathrm{Tr}\left(X_p^T X_p P\right) + \frac{1}{n_u} \mathrm{Tr}\left(X_u^T X_u P\right) \tag{21}$$

$$= \mathrm{Tr}\left(D X^T X P\right) \tag{22}$$

Where D is a diagonal matrix for which the first d_p diagonal entries are n_p^{-1} and the remaining diagonal entries are n_u^{-1}.

For column-normalized PCA, we take the vanilla PCA components of $XD^{1/2}$. Thus, column-normalized PCA is equivalent to maximizing $\mathrm{tr}\left(D^{1/2} X^T X D^{1/2} P\right)$. We observe that because of Assumption C, $X^T X$ is block diagonal and can be decomposed as $X_p^T X_p + X_u^T X_u$. Now we can write:

$$\mathrm{tr}\left(D^{1/2} X^T X D^{1/2} P\right) \tag{23}$$

$$= \mathrm{tr}\left(D^{1/2} \left(X_p^T X_p + X_u^T X_u\right) D^{1/2} P\right) \tag{24}$$

$$= \mathrm{tr}\left(D^{1/2} \left(X_p^T X_p\right) D^{1/2} + D^{1/2} \left(X_u^T X_u\right) D^{1/2} P\right) \tag{25}$$

Because of Assumption D, the last line can be written as:

$$\mathrm{tr}\left(D^{1/2} \left(X_p^T X_p\right) D^{1/2} + D^{1/2} \left(X_u^T X_u\right) D^{1/2} P\right) \tag{26}$$

$$= \mathrm{tr}\left(D \left(X_p^T X_p\right) + D \left(X_u^T X_u\right) P\right) \tag{27}$$

$$= \mathrm{Tr}\left(D X^T X P\right) \tag{28}$$

□

A.3.3 Theorem 5.

Proof. Observe that the objective for *Item-Weighted PCA* can be re-written as:

$$\sum_{j=1}^{d} w_j \left\langle S_{\cdot j}, \widehat{X}_{\cdot j} \right\rangle = \sum_{j=1}^{d} w_j \left\langle X_{\cdot j}, \widehat{X}_{\cdot j} \right\rangle \tag{29}$$

$$= \langle XD, XP \rangle \tag{30}$$

$$= \mathrm{Tr}\left(D X^T X P\right) \tag{31}$$

Where D is a diagonal matrix and entry $D_{jj} = w_j$. Thus, the two baselines and *Interpolate-Item-Weighted PCA* can we written in terms of Equation 31 with varying definitions of D.

Observe that the only difference between Equation 31 and the standard PCA objective is the addition of the weight matrix D. Now, we leverage Assumption C and D to show that the weight matrix D does not change the principal components but only their order. To see this, let V be the eigenvectors of X^TX. We can write $DV = VD$ because for all entries i, j such that $V_{ij} > 0$, $D_{ii} = D_{jj}$. Thus, the objective for *Item-Weighted PCA* becomes:

$$\text{Tr}\left(DX^TXP\right) = \text{Tr}\left(D\left(V\Sigma V^T\right)P\right) \tag{32}$$

$$= \text{Tr}\left(\left(V\left(D\Sigma\right)V^T\right)P\right) \tag{33}$$

$$\tag{34}$$

Σ is the diagonal matrix of eigenvalues. We can now see that the eigenvectors are still V but the eigenvalues are now scaled to $D\Sigma$. Furthermore, the eigenvectors of X^TX are $\{V_p, V_u\}$ given that X^TX is block diagonal. In the below, let λ_i^u be the i^{th} largest eigenvector of $X_u^TX_u$ and λ_i^p be the same for $X_p^TX_p$.

We can bound the sum of eigenvalues of $X_p^TX_p$ as follows:

$$\sum_{i=1}^{r}\lambda_i^p = \text{Tr}\left(X_p^TX_p\right) \tag{35}$$

$$= \|X_p\|_F^2 \tag{36}$$

$$= n_p d_p \tag{37}$$

Analogous steps show that the sum of eigenvalues of $X_u^TX_u$ equals $n_u d_u$. Now, we use Assumption E to establish the ratio between the leading eigenvalues of the group covariance matrices:

$$\frac{\lambda_1^p\left(\sum_{i=1}^{r}\beta^{-i+1}\right)}{\lambda_1^u\left(\sum_{i=1}^{r}\beta^{-i+1}\right)} = \frac{n_p d_p}{n_u d_u} \tag{38}$$

$$\frac{\lambda_1^p}{\lambda_1^u} = \frac{n_p}{n_u}\frac{d_p}{d_u} \tag{39}$$

$$\frac{\lambda_1^p}{\lambda_1^u} = \sqrt{\frac{n_p}{n_u}} \tag{40}$$

$$\tag{41}$$

Thus, all eigenvalues of $X_p^TX_p$ are $\sqrt{\frac{n_p}{n_u}}$ times larger than the corresponding eigenvalue for $X_u^TX_u$.

In the case of Vanilla *PCA*, $D = I$. We can show that the largest eigenvalue of $X_u^TX_u$ is still smaller than the smallest non-zero eigenvalue of $X_p^TX_p$:

$$\lambda_r^p = \lambda_1^p\beta^{1-r} \tag{42}$$

$$= \lambda_1^u\left(\sqrt{\frac{n_p}{n_u}}\right)\beta^{1-r} \tag{43}$$

$$\geq \lambda_1^u\beta^{r-1}\beta^{1-r} \tag{44}$$

$$\geq \lambda_1^u \tag{45}$$

Thus the r largest eigenvalues correspond with V_p.

On the other hand, we can show that when $D_{ii} = n_p^{-1}\ \forall i \in I_p$ and $D_{ii} = n_u^{-1}\ \forall i \in I_u$, as in the case of column-normalized PCA, the smallest re-scaled eigenvalue for $X_u^TX_u$ will be larger than the largest re-scaled eigenvalue of $X_p^TX_p$

$$\lambda_r^u n_u^{-1} = \lambda_1^u\beta^{r-1}n_u^{-1} \tag{46}$$

$$= \lambda_1^p\sqrt{\frac{n_u}{n_p}}\beta^{1-r}n_u^{-1} \tag{47}$$

$$= \frac{\lambda_1^p}{n_p}\sqrt{\frac{n_p}{n_u}}\beta^{1-r} \tag{48}$$

$$\geq \frac{\lambda_1^p}{n_p}\beta^{r-1}\beta^{1-r} \tag{49}$$

$$\geq \frac{\lambda_1^p}{n_p} \tag{50}$$

Thus, after re-scaling with column-normalized PCA all of the top r eigenvectors will correspond with V_u.

In the case of *Interpolate-Item-Weighted PCA*, the rescaled i^{th} eigenvalue for $X_p^TX_p$ will be:

$$\frac{1}{\sqrt{n_p}}\lambda_i^p = \frac{1}{\sqrt{n_p}}\frac{\sqrt{n_p}}{\sqrt{n_u}}\lambda_i^u \tag{51}$$

$$= \frac{1}{\sqrt{n_u}}\lambda_i^u \tag{52}$$

Thus, the re-scaled eigenvalues for $X_p^TX_p$ exactly equal the rescaled eigenvalues for $X_u^TX_u$. In taking the top r eigenvectors, the final set will contain one half from V_p and another half from V_u. □

A.4 Additional Proofs

Theorem 8. *Let $X \in \mathbb{R}^{n\times d}$, then the i^{th} principal component reduces the reconstruction error by:*

$$\|X - XU_{i-1}U_{i-1}^T\|_F^2 - \|X - XU_iU_i^T\|_F^2 = \sigma_i^2$$

Where the columns of U_i are the leading i principal components and σ_i is the i^{th} largest singular value of X by magnitude.

Proof. The reconstruction error f for a given projection matrix $P = UU^T$ can be re-written as:

$$\begin{aligned}
f(P) &= \|X - XP\|_F^2 \\
&= \text{tr}\left((X - XP)^T(X - XP)\right) \\
&= \text{tr}\left(X^TX - X^TXP - PX^TX + PX^TXP\right) \\
&= \text{tr}\left(X^TX\right) - \text{tr}\left(X^TXP\right) - \text{tr}\left(PX^T\right) + \text{tr}\left(PX^TXP\right) \\
&= \text{tr}\left(X^TX\right) - \text{tr}\left(X^TXP\right) - \text{tr}\left(X^TP\right) + \text{tr}\left(X^TXPP\right) \\
&= \text{tr}\left(X^TX\right) - \text{tr}\left(X^TXP\right) - \text{tr}\left(X^TP\right) + \text{tr}\left(X^TXP\right) \\
&= \text{tr}\left(X^TX\right) - \text{tr}\left(X^TXP\right) \\
&= \text{tr}\left(X^TX\right) - \text{tr}\left(U^TX^TXU\right)
\end{aligned}$$

Vanilla PCA minimizes reconstruction error which is equivalent to maximizing $\text{tr}\left(U^TX^TXU\right)$. The matrix X^T is a symmetric matrix that can be diagonalized as $V\Sigma^2V^T$ where the columns of V are the

right singular vectors of X and Σ is a diagonal matrix where the diagonal values are the singular values of X sorted by magnitude.

Maximizing $\text{tr}\left(U^T X^T X U\right)$, where the columns of U are orthonormal then amounts to setting the columns of U to be the leading right singular vectors of X. Now the reduction in reconstruction error can be written as:

$$\begin{aligned} f\left(P_{i-1}\right) - f\left(P_i\right) &= \text{tr}\left(U_i^T X^T X U_i\right) - \text{tr}\left(U_{i-1}^T X^T X U_{i-1}\right) \\ &= \text{tr}\left(V_i^T \left(V \Sigma^2 V^T\right) V_i\right) - \text{tr}\left(V_{i-1}^T \left(V \Sigma^2 V^T\right) V_{i-1}\right) \\ &= \sum_{j=1}^{i} \sigma_j^2 - \sum_{j=1}^{i-1} \sigma_j^2 \\ &= \sigma_i^2 \end{aligned}$$

□

B SUPPLEMENTAL FIGURES

B.1 Singular value scaling of Bernoulli matrices

We empirically check that Assumption A is satisfied for the class of Bernoulli matrices used in Theorem 1.

We fix $M = 20$, and vary n from 100 to 100,000. For each item $j \in [M]$, we draw a random number $p_j \in [0.1, 1]$ uniformly, which represents the Bernoulli parameter for item j. Then, we draw a matrix $X \in \{0,1\}^{n \times M}$, where $X_{ij} \sim \text{Bernoulli}(p_j)$ independently for each i, j. Then, we denote by $s^2_{\min}(X)$ to be the smallest value of the squared singular values of X. For Assumption A to be satisfied, it must be that $s^2_{\min}(X)$ should scale linearly in n.

For one set of values of the Bernoulli parameters $\{p_j\}_{j=1,\ldots,M}$, we draw the random matrix X 500 times, and we show the average $s^2_{\min}(X)$ value as well as the 99'th percentile values in Figure 6. The figure shows that the smallest squared singular value does indeed increase linearly with high probability.

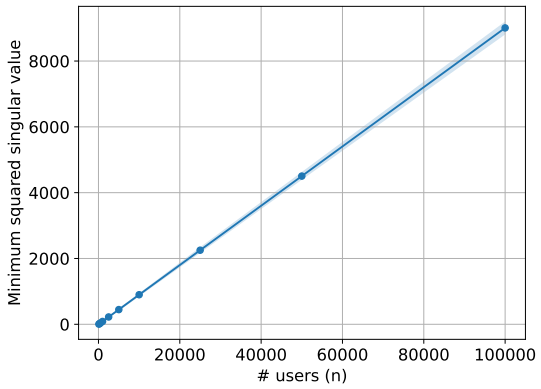
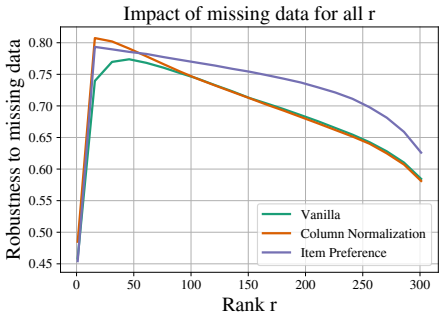
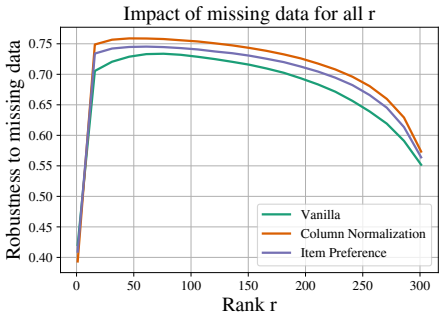


Figure 6: The line corresponds to the average of the smallest squared singular value of the random Bernoulli matrix X. The shaded region corresponds to the 1 and 99th percentile values.



(a) LastFM



(b) MovieLens

Figure 7: We present robustness results introduced in Figure 5 for all values of r. For a given value of r, we summarize the algorithm's robustness by averaging the Item AUC-ROC over all values of α.

B.2 Robustness results for all values of r

To supplement Figure 5, we include robustness results for all values of r in Figure 7. For a given value of r, we summarize each algorithm's performance by taking the average AUC-ROC for all values of α. Figure 7 shows that for the LastFM dataset, our algorithm outperforms both baselines for $r > 50$, not only $r = 100$; also, for MovieLens, our algorithm performs comparably with the baselines for all values of r.