Folding: Why Good Models Sometimes Make Spurious Recommendations

Doris Xin[*]
University of Illinois at
Urbana-Champaign
201 N Goodwin Ave
Urbana, Illinois 61801
dorx0@illinois.edu

Nicolas Mayoraz
Google Research
1600 Amphitheatre Parkway
Mountain View, California 94043
nmayoraz@google.com

Hubert Pham
Google Research
1600 Amphitheatre Parkway
Mountain View, California 94043
hubertpham@google.com

Karthik Lakshmanan
Google Research
1600 Amphitheatre Parkway
Mountain View, California 94043
lakshmanan@google.com

John R. Anderson
Google Research
1600 Amphitheatre Parkway
Mountain View, California 94043
janders@google.com

ABSTRACT

In recommender systems based on low-rank factorization of a partially observed user-item matrix, a common phenomenon that plagues many otherwise effective models is the interleaving of good and spurious recommendations in the top-K results. A single spurious recommendation can dramatically impact the perceived quality of a recommender system. Spurious recommendations do not result in serendipitous discoveries but rather cognitive dissonance. In this work, we investigate *folding*, a major contributing factor to spurious recommendations. Folding refers to the unintentional overlap of disparate groups of users and items in the low-rank embedding vector space, induced by improper handling of missing data. We formally define a metric that quantifies the severity of folding in a trained system, to assist in diagnosing its potential to make inappropriate recommendations. The *folding metric* complements existing information retrieval metrics that focus on the number of good recommendations and their ranks but ignore the impact of undesired recommendations. We motivate the folding metric definition on synthetic data and evaluate its effectiveness on both synthetic and real world datasets. In studying the relationship between the folding metric and other characteristics of recommender systems, we observe that optimizing for goodness metrics can lead to high folding and thus more spurious recommendations.

KEYWORDS

collaborative filtering; matrix factorization; folding; evaluation metric; MNAR

1 INTRODUCTION

Recommender systems based on collaborative filtering propose items to users by generalizing from sparse observed user-item affinities. Prior to live deployment, system developers measure the quality of recommenders using offline metrics that typically treat the recommendation problem as a regression or ranking problem. Regression metrics (e.g., RMSE, MAE) measure how accurately the system can predict the affinity of some hold-out user-item affinities, whereas ranking metrics (e.g., Precision@K, MAP) measure how well hold-out observations intersect with predicted recommendations, for each user. Such metrics are one-sided: they are only *goodness* metrics, focused on measuring recommendation performance based on observed data, while ignoring the possible presence and range of inappropriate recommendations.

In the development of production recommendation systems, disregarding bad recommendations is a critical issue, as a single inappropriate recommendation, even surrounded by many excellent ones, can have a disastrous impact on user experience. Furthermore, not all bad recommendations are equally undesirable. For example, when recommending movies to a user that mostly watches animated Disney movies, a recommendation set that consists of relevant animated videos, along with a PG romantic comedy, is perhaps questionable. Arguably far worse are recommendations with those same animated movies but alongside an NC-17 horror movie. This issue is exacerbated in cold-start situations, where the model makes predictions for new items when much of the interaction data is unobserved.

The above scenarios arise because observed data is not uniformly distributed among user-item pairs. In most applications, we do not have explicit ratings from clearly unrelated pairs, e.g., between a child viewer and an horror movie. Hence, classical goodness metrics focused on evaluation against observed data cannot effectively detect and differentiate the scenarios above. This problem is related to the issue of data *missing not at random* (MNAR), well studied in the recommendation system literature [19]. Proposed solutions so far include building models that are explicitly aware of bias in the training data [27] and focusing on the recommendations at the top [2], yielding models that are less likely to score inappropriate recommendations highly. However, to the best of our knowledge,

[*] Work done while the author was at Google Research.

RecSys'17, August 27–31, 2017, Como, Italy.
© 2017 Copyright held by the owner/author(s). ISBN 978-1-4503-4652-8/17/08.
DOI: http://dx.doi.org/10.1145/3109859.3109911

there has been little attention devoted to designing a *badness* metric that captures the range of inappropriate recommendations. The lack of labeled data for bad user-item pairs explains the inherent complexity in designing a badness metric, and identifying specific inappropriate user-item pairs for such a metric is no simpler than solving the original regression problem.

Spurious estimations are predictions that are not well-supported by the observations from which the model is trained. Thus, they are artifacts of the model rather than being inherent to the data. It is important to note that these are different from diverse recommendations, where the recommendations are explicitly aimed at satisfying a diversity objective that leads to serendipitous discoveries. By contrast, spurious estimations are made by the recommender model due to lack of associated data. Semantically, spurious recommendation could mean that an item is inappropriate for user (e.g., horror films to children).

We propose the *folding metric* as a measure of badness with respect to unobserved data in a recommendation model. The metric pursues an approach based on inferring the geometry of the embedding space that results from matrix factorization. In the matrix factorization setting, users and items are embedded in a vector space such that elements with high affinity are near in the space by some distance metric, and dissimilar items are far apart. However, as we will demonstrate, when missing data is not handled properly, the resultant space can embed similar users and items to be well-positioned with respect to each other, but place or *fold* those elements in the same region as another unrelated group of coherent users and items. Two groups are unrelated if their members have never interacted with each other. When generating recommendations for a user by returning its nearest-neighbors in the embedding space, folding leads to spurious recommendations. Using synthetic data, we illustrate how the folding metric complements a goodness metric. Optimizing solely for goodness can lead to models with high folding, which in turn translates to more inappropriate recommendations.

In our experiments, we evaluate the folding metric on the MovieLens dataset [7] and empirically show that it is robust to different estimations of the likelihood of user-item interactions, which is a prerequisite to the folding metric.

2 PRELIMINARIES

A set of m users and n items for which we have some observed ratings $a_{ij} \in \mathbb{R}$ constitutes the setting of a regression problem in collaborative filtering. The observed ratings a_{ij} partially define a matrix $\boldsymbol{A} \in \mathbb{R}^{m \times n}$, where $\Omega \subset \{1, \ldots, m\} \times \{1, \ldots, n\}$ denotes the set of observed entries of $\boldsymbol{A}$, and typically $|\Omega| \ll mn$. These observations can be explicit user-item ratings, or they can reflect the number of exposures and interactions (e.g., click through rate).

The goal of a recommender system is to estimate which unobserved entries have the largest latent values in any given row $\boldsymbol{a}_i$ of $\boldsymbol{A}$. A popular approach is to assume that the unknown underlying process generating observations defines a low-rank k matrix. The goal is then to approximate the partially defined matrix $\boldsymbol{A}$ with a low-rank matrix $\boldsymbol{U}^\top \boldsymbol{V}$ where $\boldsymbol{U} \in \mathbb{R}^{k \times m}, \boldsymbol{V} \in \mathbb{R}^{k \times n}$, by solving the following optimization problem:

$$\min_{\boldsymbol{U},\boldsymbol{V}} \ \|\boldsymbol{W} \odot (\boldsymbol{A} - \boldsymbol{U}^\top \boldsymbol{V})\|_F^2 \ + \ \lambda(\|\boldsymbol{U}\|_F^2 \ + \ \|\boldsymbol{V}\|_F^2), \tag{1}$$

where $\boldsymbol{W} \in \mathbb{R}_+^{m \times n}$ is a weight matrix, $\odot$ denotes the Hadamard matrix product, $\| \cdot \|_F$ represents the Frobenius norm, and $\lambda \geq 0$ is a regularization parameter.

The problem in Equation (1) is well defined if for each $(i,j) \notin \Omega$, either $w_{ij} = 0$, or a_{ij} is predetermined. Handling missing data by selecting the appropriate w_{ij} and a_{ij} for $(i,j) \notin \Omega$ is of crucial importance to any factorization-based recommendation algorithm. In regression problems, unobserved entries are set to a prior. Common choices for the prior are a global constant and functions of i and/or j. For example, in a user-movie rating scenario, users have mostly watched and rated movies they like, hence the prior for all ratings should be lower than the global average observed ratings.

2.1 Weighting the Unknown

Various propositions for $\boldsymbol{W}$ have led to a broad range of algorithms. On one end of the spectrum, setting $\boldsymbol{W} = \boldsymbol{1}$ and $a_{ij} = 0, \forall (i,j) \notin \Omega$ in Equation (1) converts the problem to an SVD decomposition of $\boldsymbol{A}$ (under $\lambda = 0$). This approach is known to lead to recommendations that can be quite poor when $\boldsymbol{A}$ is very sparse, because the loss function is dominated by the unobserved data. In practice, a sparsity rate of 10^{-3} or lower is common, thus making this approach inapplicable in most settings. On the other end of the spectrum, the unobserved data can be ignored all together by setting

$$\boldsymbol{W} \ = \ \boldsymbol{1}^\Omega,$$

where $\boldsymbol{1}^\Omega$ is the indicator matrix of the set Ω such that $\boldsymbol{1}_{ij}^\Omega = 1$ for $(i,j) \in \Omega$ and 0 otherwise. As we will demonstrate in Section 4, this can lead to spurious recommendations.

[12] shows that attributing a small uniform weight to all unobserved data leads to success on the Netflix Prize, by setting

$$\boldsymbol{W} \ = \ \boldsymbol{1}^\Omega + \alpha \boldsymbol{W}^0 \ , \tag{2}$$

where $\alpha > 0$ is a meta-parameter that controls the importance attributed to the unobserved user-item pairs. This meta-parameter α needs careful tuning. A reasonable choice for α is in the order of the sparsity rate of $\boldsymbol{A}$, thereby sharing evenly the total weight of the observed and unobserved data in Equation (1). Different choices for $\boldsymbol{W}^0 \in \mathbb{R}^{m \times n}$ have been studied in [12], where $\boldsymbol{W}^0$ is the all-ones matrix; in [24], where $\boldsymbol{W}^0$ is rank 1; and in [25], where $w_{ij} = c_i c_j$ with multiple options for c_i and c_j.

2.2 Optimization Strategies

The optimization problem in Equation (1) has been widely studied in the literature in the context of collaborating filtering and recommender systems, and the proposed solutions broadly fall into two categories: weighted alternating least square solvers (WALS) and stochastic gradient descent solvers (SGD).

WALS presents the considerable advantage of taking into account the full matrix $\boldsymbol{A}$, down-weighting appropriately the unobserved entries, often leading to more accurate models [30]. According to [12] and [24], as long as the weight matrix is of the simple form of Equation (2), WALS solvers are able to handle large matrices.

The main advantage of SGD is flexibility. For example, SGD naturally supports (i) various cost functions beyond squared loss (e.g. logistic loss, SoftMax + cross-entropy), (ii) non-linearity (e.g. in the form of hidden layers of a deep neural network), and (iii) integration of meta data (e.g. user geo-location or movie language or genre). However, the efficiency of SGD relies heavily on negative sampling (selecting for each batch some $(i, j) \notin \Omega$ and setting $w_{ij} > 0$), and many sampling strategies have been proposed [13, 23]. For this reason, SGD algorithms become slow to converge for very large and extremely sparse problems, and the largest problems that can be solved by SGD in reasonable time are orders of magnitude smaller than the sizes mentioned above handled by WALS.

We have done extensive comparisons between WALS and SGD in other contexts, and for appropriate negative sampling strategies aimed at emulating the same weight of unobserved data α, the optimal solution reached by SGD is qualitatively the same as the one obtained by WALS, with the latter being typically much faster. Therefore, in all our experiments in this paper, we use a WALS solver to optimize Equation (1).

3 FOLDING

3.1 The Folding Phenomenon: A Case Study

To establish some intuition on the folding problem, consider the toy example user-movie ratings matrix $\boldsymbol{A}$ shown in Figure 1. Rows of $\boldsymbol{A}$ represent users while columns denote movies. Users rate movies on a scale of 1 to 5 with 5 being the most favorable. Whitespace in the matrix indicates unobserved ratings.

Matrix factorization based techniques assume that the user-rating pairs are generated from a low-rank process. In this example we explicitly define the underlying process in terms of user and item groups, visualized by the block structure in Figure 1(a). Note that this block structure can be extracted from real datasets by rearranging the rows and columns to place similar rows/columns next to each other. Movies are grouped by either their genres or language, while users are grouped by the genres of movies they've watched, with the exception that i) some popular Family/Action/Romantic films have been rated by users in more than one block and ii) only bilingual users have rated Japanese films.

Figure 1(b) depicts $\boldsymbol{A}$, the matrix containing observed ratings for some user-movie pairs. Users are more likely to rate movies they enjoy than not, echoing the same observed trend in the MovieLens dataset. The optimization problem in Equation (1) is solved for $\boldsymbol{A}^o$, the centered version of $\boldsymbol{A}$ (the mean rating is subtracted), using WALS with $a^o_{ij} = 0 \; \forall (i, j) \notin \Omega$, embedding dimension $k = 3$, and weight on the unobserved $\alpha = 0$. Figure 1(c) shows $\boldsymbol{A}^* = \boldsymbol{U}^\top \boldsymbol{V}$, the approximation of $\boldsymbol{A}^o$ using the WALS solutions. Let $\bar{\boldsymbol{B}}$ be the masking matrix that has 0 for the dark blocks in Figure 1(a) and 1 everywhere else. Figure 1(d) shows $(\boldsymbol{A}^* - \boldsymbol{A}^o) \odot \bar{\boldsymbol{B}}$, the difference between the predicted ratings and the actual ratings for the blocks without user-movie interactions. The large positive errors are due to folding.

Figure 2 visualizes the embedding vectors to provide insight into the embedding space geometry underlying the folding phenomenon. It shows a 2D projection of the 3D WALS embedding vectors in $\boldsymbol{U}$ and $\boldsymbol{V}$. For visual clarity, we only show the subset of user groups that are most impacted by folding. The embedding vectors exhibit a few undesired characteristics: i) embedding vectors for horror films are close to embedding vectors for children, who have rated nothing but family films; ii) action fans are placed in closer proximity to Japanese films than action films, when none have previously rated any Japanese films. We refer to this phenomenon in which disparate groups are collapsed onto the same region in the embedding space as folding.

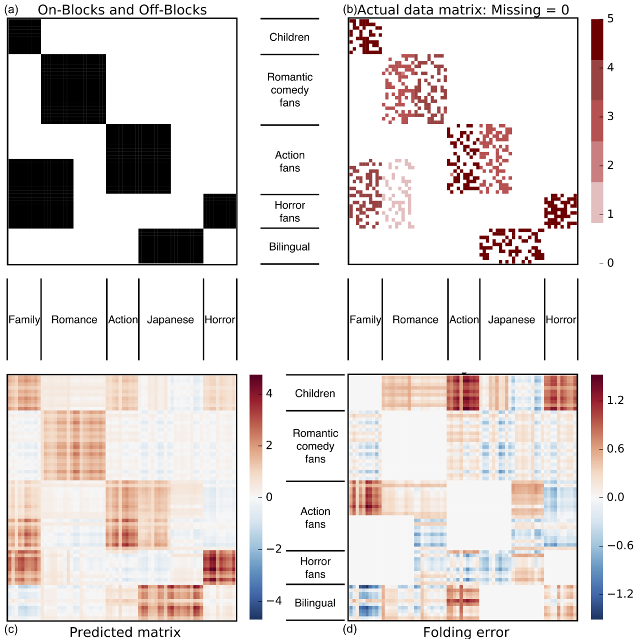

Figure 1: Synthetic User-Movie Ratings.

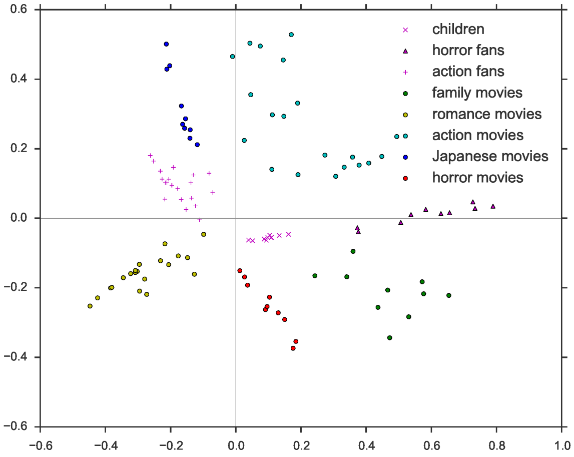

Figure 2: Embedding Vectors for User and Movies in Figure 1

In a typical recommender system powered by low-rank matrix factorization, the standard method for generating recommendations, after the embedding vectors are trained, is to take the top-K items, excluding previously rated items, with embedding vectors that achieve the highest cosine similarity (or dot product) with a given user vector. The predicted matrix shown in Figure 1(c) illustrates the ramifications: for children, while the family movie group is recovered correctly, other groups of movies are ranked too high (action, horror) and some of these movies are bound to creep in the top-K recommendations.

Recommending an uninteresting family film to a child is far less of an offense than recommending a horror film. Determining the impact of a bad recommendation is hinged upon our ability to discern why a user-item rating is missing from the observation set Ω. The Missing Not At Random (MNAR) principle postulates that not all user-item pairs have equal likelihood of being unobserved. As shown in Figure 1, the likelihood is tied to the consumption pattern – for a given user, a movie in a genre she has regularly rated is much less likely to have a missing rating than in a genre she has never interacted with before. We capture the missing data mechanism via *relatedness*, to be discussed in Section 3.4.

3.2 An Example of Folding using MovieLens

A concrete example of folding is shown in Table 1 using the MovieLens data [7]. Treating the recommendation problem as a regression problem with a_{ij} being the star rating user i gave to movie j, we solve the optimization problem in Equation (1) twice, once ignoring all missing data (setting $\alpha = 0$ in Equation (2)), and once using $\alpha = 0.001$, and $a_{ij} = 2\ \forall (i,j) \notin \Omega$). The MovieLens data is first partitioned into training and hold-out, and Equation (1) is optimized using the training data alone.

According to the mean squared error (MSE) measured on the hold-out set, the solution obtained while ignoring the missing data is quantitatively better than the one using the missing data. However, a cosine-based nearest neighbors analysis clearly reveals folding in the first case, indicated by high-ranking inappropriate recommendations reported in Table 1. The query movie "Cinderella" is a musical targeting children and youth. The second set, composed predominantly of comedies and adolescent drama, is thus more fitting than the crime thrillers and drama for a mature audience in the first set, even though the cosine measures are higher in the first set.

3.3 Causes of Folding

Folding is best explained by partitioning the unobserved data into related and unrelated missing values. A related user-item pair is unobserved for potentially multiple reasons. For example, the user might not interact with an item due to lack of exposure but would otherwise have an opinion about that item, or perhaps the item is extremely popular and the missing observation denotes implicit negative affinity. Together with Ω, the set of observed user-item pairs, the missing related data form the related user-item pairs. Each related missing user-item pair can take any rating value. In Figure 1, the related missing user-item pairs are the missing entries in the populated blocks of the data matrix in (b). Serendipitous recommendations are always about unobserved related data.

<table>
<tr><th colspan="2">Recs Ignoring Missing Data</th><th colspan="2">Recs Using Missing Data</th></tr>
<tr><th>Movie</th><th>Cosine</th><th>Movie</th><th>Cosine</th></tr>
<tr><td><b>Cinderella (1997)</b>
Children|Fantasy|Musical|Romance</td><td>1</td><td><b>Cinderella (1997)</b>
Children|Fantasy|Musical|Romance</td><td>1</td></tr>
<tr><td>Ransom (a.k.a. The Terrorists) (1975)
Crime|Thriller</td><td>0.912</td><td>27 Dresses (2008)
Comedy|Romance</td><td>0.821</td></tr>
<tr><td>Memoirs of a Geisha (2005)
Drama|Romance</td><td>0.909</td><td>Twilight (2008)
Drama|Fantasy|Romance|Thriller</td><td>0.810</td></tr>
<tr><td>Hairspray (2007)
Comedy|Drama|Musical</td><td>0.897</td><td>Made of Honor (2008)
Comedy|Romance</td><td>0.780</td></tr>
<tr><td>Monster House (2006)
Adventure|Animation|Children|
Comedy|Drama|Fantasy|Mystery</td><td>0.895</td><td>High School Musical 2 (2007)
Comedy|Drama|
Musical|Romance</td><td>0.771</td></tr>
<tr><td>Cold Mountain (2003)
Drama|Romance|War</td><td>0.892</td><td>House Bunny, The (2008)
Comedy</td><td>0.764</td></tr>
<tr><td>Millions (2004)
Children|Comedy|Crime|Drama|Fantasy</td><td>0.890</td><td>Step Up (2006)
Drama|Romance</td><td>0.761</td></tr>
<tr><td>Seven Pounds (2008)
Drama</td><td>0.890</td><td>P.S. I Love You (2007)
Comedy|Drama|Romance</td><td>0.761</td></tr>
<tr><td>Tuesdays with Morrie (1999)
Drama</td><td>0.889</td><td>Fool's Gold (2008)
Action|Adventure|Comedy|Romance</td><td>0.738</td></tr>
<tr><td>Bobby (2006)
Drama</td><td>0.888</td><td>Nanny Diaries, The (2007)
Comedy|Drama|Romance</td><td>0.735</td></tr>
</table>

Table 1: Top 9 recommendations for the movie Cinderella according to a two different models.

In contrast, an unrelated missing user-item pair characterizes a user who has no interest in an item (e.g., children and horror films or Japanese films and non-speakers). Unrelated missing data often constitute the vast majority of the missing observations, due to some inherent structure in the data generation process: user-item consumption is naturally limited by region, languages, or it is artificially restricted by targeting. In the example of Section 3.1, the unrelated missing observations are all the user-item pairs outside of the dark blocks in Figure 1(a).

Folding pertains to falsely assigning high affinity to unrelated missing data. By contrast, identifying similarities in related missing data is considered generalization by the model from known data. When a model folds, it leads to spurious recommendations, often inter-leaved with good recommendations, which makes the issue hard to detect. The likelihood of a model to fold is exacerbated by the fact that the indicator matrix $\mathbf{1}^{\Omega}$ is often close to a block sparse matrix (as in Figure 1), due to implicit partitioning of users and items based on metadata such as countries, languages, interests, etc— most viewers will watch movies with either the sound track or subtitles in a language they understand; many music listeners are interested in 2-3 genres of music; online shoppers are shown ads about items from a small pool of topics targeted to them.

To illustrate how such situation can lead to folding, consider the simplified scenario where $\mathbf{1}^{\Omega}$ is K-block-diagonal, inducing a partition of users into K populations and items into K groups. When unobserved data are ignored ($\alpha = 0$), there are no constraints between users in population k and items in group $l \neq k$. Hence, Equation 1 can split into K independent sub-problems. In any optimal solution the embedding vectors of users and items of a population k are organized in specific positions with respect to each-other, but they can fold on top of users and items of population $l \neq k$.

3.4 Relatedness

The folding metric is based upon the notion of *relatedness* between a user i and an item j, which captures the likelihood of interaction between i and j, regardless of the rating. Pairwise relatedness between users and items can be modeled in a matrix $\boldsymbol{R} \in [0,1]^{m \times n}$. Such a matrix is usually not available and needs to be estimated first. Section 4 explores and evaluates one way to estimate $\boldsymbol{R}$ for the specific case of MovieLens data, along with a generic approach, which is described next.

One can compute a low-rank approximation of the binary matrix $\mathbf{1}^{\Omega}$, and then apply a coefficient-wise monotonic transformation (e.g. translation + normalization) to get $\boldsymbol{R}$ in $[0,1]^{m \times n}$. This approximation can be done using WALS with a high unobserved weight parameter α (e.g. $\alpha = 1$), or simply using SVD. The rank should ideally be just large enough to capture the underlying hidden structure of $\mathbf{1}^{\Omega}$, but small enough not to encode any noise. This low-rank approximation is a form of smoothing of $\mathbf{1}^{\Omega}$ with the desired property that for a majority of user-item pairs, r_{ij} is large if $(i,j) \in \Omega$ and small otherwise. However, r_{ij} can also be somewhat large if user i and item j are not directly connected, i.e., $(i,j) \notin \Omega$, but closely related in the bipartite graph defined by Ω. Conversely, this approach also allows for cases where r_{ij} is small for $(i,j) \in \Omega$, which is a desirable property in the case of noisy data.

3.5 The Folding Metric Definition

Definition 3.1. Given a measure of relatedness $\boldsymbol{R}$, the folding metric F_R, which assesses the propensity of a user-item similarity measure $\boldsymbol{S}$ to cause spurious recommendations, is defined as

$$F_R(\boldsymbol{S}) \;=\; \frac{1}{mn} \sum_{i,j} \max(0, s_{ij} - r_{ij})$$

The asymmetry due to $\max(0, \cdot)$ causes the folding metric to be only a badness measure: folding increases when unrelated pairs are considered similar but is not affected when related pairs are rated as dissimilar. First, related pairs are taken care of by goodness metrics (which are always used jointly with the folding metric), as they are likely supported by observed ratings. Second, predicting low ratings on related pairs is legitimate and does not contribute to folding.

In the context of matrix factorization, the similarity measure $\boldsymbol{S}$ is defined over the factors $(\boldsymbol{U}, \boldsymbol{V})$. Its exact definition can vary based on the application, but a reasonable choice is the cosine similarity between user vector $\boldsymbol{u}_i$ and an item vector $\boldsymbol{v}_j$. Note that given Definition (3.1), (i,j) pairs with negative s_{ij} make no contribution to the folding metric, regardless of the value for $r_{ij} \in [0,1]$.

The folding metrics $F_R(\boldsymbol{S})$ and $F_R(\boldsymbol{S}')$ for different similarity measures $\boldsymbol{S}$ and $\boldsymbol{S}'$ are comparable only if $\boldsymbol{S}$ and $\boldsymbol{S}'$ have similar distributions. For example, the folding value of a cosine similarity and the folding value of a dot-product similarity are both interesting in their own rights, but cannot be compared.

3.6 Relationship to Other Metrics

In this section we discuss the relationship between the folding metric and existing metrics for evaluating recommender systems. Below, let Ω_i^R denote the set of related items to user i, and let Ω_i^U denote the set of unrelated items to user i.

3.6.1 *Precision.*

Precision is applicable only in the context of recommending good items, a task with a binary objective. During offline evaluation for this task, a fraction of the items rated favorably by the user is held out from the training set, and the model is measured on its ability to retrieve the hold-out set. Traditionally, missing observations are treated as negatives in such evaluations, which require negative labels. That is, recommending an item not in the hold-out set is considered a false positive. Since most applications only recommend to the users a small fraction of all available items, precision is commonly measured on only the top-K recommendations, in a variant known as precision@K,

$$\text{Precision@}K = \frac{TP}{K} = 1 - \frac{FP}{K}$$

where TP is the number of true positives and FP is the number of false positives corresponding to the number of missing observations in the standard treatment.

While the folding metric operates mainly in a space orthogonal to the focus of precision, high folding can negatively impact performance measured by precision@K. By the MNAR principle, $E[P_{obs}(i,j)] \gg E[P_{obs}(i,k)]$ for movie j in the related set Ω_i^R vs. movie k in the unrelated set Ω_i^U, where $P_{obs}(u,v)$ denotes the probability of observing a rating for movie v by user u. Let $\mathbf{y}_i$ be a recommendation set consisting of K items and $P_{FP}(i,j)$ denote the likelihood of user-item pair (i,j) being a false positive. Since all missings are regarded as negatives by precision, $E[P_{FP}(i,j)] \propto E[1 - P_{obs}(i,j)]$, which implies

$$E[\text{Precision@}K] \propto \sum_{y \in \mathbf{y}_i} E[P_{obs}(i,y)]$$

For two recommendations sets $\mathbf{y}_i$ and $\mathbf{y}_i'$ of size K such that $|\mathbf{y}_i \cap \Omega_i^U| > |\mathbf{y}_i' \cap \Omega_i^U|$, i.e. $\mathbf{y}_i$ suffers more from folding,

$$\sum_{y \in \mathbf{y}_i} E[P_{obs}(i,y)] < \sum_{y \in \mathbf{y}_i'} E[P_{obs}(i,y)] \; ,$$

implying that $\mathbf{y}_i$ with higher folding has lower precision@K in expectation. However, for a given value of precision, folding can take on a range of values because precision does not distinguish between related and unrelated missings. Folding is more severe with top-K results containing unrelated missings than related missings.

3.6.2 *Ranking Based.*

One major drawback to precision/recall is that they are not sensitive to the position of the true positives in the ranked list for the top-K results. Ranking based metrics such as discounted cumulative gain (DCG) is designed to measure the system's ability to prioritize good recommendations over the bad ones. DCG@K, the analog to precision/recall@K, is defined as

$$DCG@K = \sum_{i=1}^{K} \frac{2^{r(i)} - 1}{\log_2(i+1)}$$

where $r(i)$ indicates the relevance of the item at position i. A common choice for r is a binary indicator for observed interaction between a user and an item. As with precision/recall, missing items have a relevance of 0, effectively acting as a negative. For recommendation sets that have the same precision, the ones that have the positives positioned towards the top of the list have higher DCG.

DCG@K is also negatively impacted by high folding. The interleaving of $g \in \Omega_i^R$ and $b \in \Omega_i^U$ pushes down the ranks of positives, resulting in a lower DCG. The MNAR assumption implies $E[r(g)] > E[r(b)]$. Since DCG also does not distinguish between related and unrelated missings, the value for the folding metric can vary significantly between models with the same DCG.

3.6.3 RMSE. Root mean squared error (RMSE) is the standard choice for evaluating recommender systems on regression tasks such as user rating prediction. While the metrics above heed some notion of negatives, RMSE does not consider them at all. For the @K metrics above, a false negative occupies a spot that could have been filled with a true positive, thus lowering the metric. For RMSE, any value can be placed on the missing entries without affecting the objective. Our experiments suggest that when constrained by embedding dimensions, systems that optimize for RMSE predict high values on Ω_i^U, which indicates higher folding, to achieve more accurate predictions for Ω_i^R.

4 EXPERIMENTS

To separate the study of folding from the estimation of the relatedness matrix $\boldsymbol{R}$, we first report experiments on synthetic data, where the relatedness is known as part of the data generation process. Then, we illustrate folding on models trained on the MovieLens dataset.

4.1 Assessing Folding on Synthetic Data

The experiments reported below involve 5000 users and 6000 items. The low-rank process that defines user consumption patterns is generated by partitioning users into 5 groups of sizes varying from 400 to 1600 and items into 10 groups of sizes from 100 to 1400. Out of the 50 blocks in {user groups} × {item groups}, a quarter of them are marked as *on-blocks* (dark blocks in Figure 1(a)), which connect *related* users and items. These on-blocks are selected randomly such that any user or item is in at least one on-block. The remaining blocks are *off-blocks*, connecting *unrelated* users and items. Each block is further divided into 2 x 2 sub-blocks, and each sub-block is assigned a random integer in $[1, 5]$ representing user-item ratings. From the low-rank process mask, 25% of the user-item pairs in the on-blocks are randomly selected as part of the observed dataset, along with 2.5% of the entries in the off-blocks (assigned random ratings) to simulate noise. 25% of the observed dataset is held out for testing, excluding noisy entries. We repeat this data generation process 100 times, and Figure 3 reports means and standard deviations among these 100 experiments. In each experiment, the $\{0, 1\}$ matrix of on/off blocks is used as the relatedness matrix $\boldsymbol{R}$, i.e., $r_{i,j} = 1$ if and only if the user i and item j pair belongs to an on-block. Each full data matrix in $\{0, \ldots, 5\}^{5000\times6000}$ is factored using WALS, with a regularization parameter $\lambda = 0.1$ while varying k, the embedding dimension, and α, the weight on unobserved user-item pairs. Figure 3 shows the RMSE measured on hold-out data and the folding metric measured on all user-item pairs for different combinations of k and α. Folding is calculated for the cosine similarity between user and item embeddings.

Note that since in these datasets $\frac{3}{4}$ of the blocks are off-blocks, $r_{i,j} = 0$ for $\frac{3}{4}$ of the user-item pairs in expectation. The maximum folding value is 0.75 when the model assigns maximum cosine similarity $s_{i,j} = 1$ to every unrelated pair. If the expected cosine similarity over the unrelated pairs is just average ($s_{i,j} = 0.5$) the folding measure is 0.375.

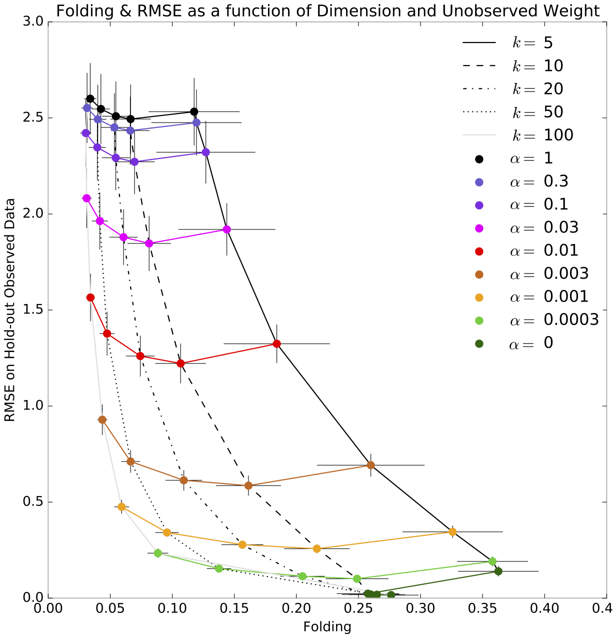

Figure 3: Negative correlation between RMSE and Folding, varying dimension k and unobserved weight α.

For any fixed embedding dimension k, decreasing the weight on unobserved data α improves the error rate on hold-out data but deteriorates the folding metric. This trend is more pronounced in lower dimensions, as the model does not have enough resolution to represent the data well. Ignoring the numerous unobserved pairs naturally leads to more folding. Similarly, fixing α and decreasing k worsens the folding metric, as the information is more compressed, but improves the error rate, as overfitting decreases, up to the point where the model underfits.

Optimizing only for RMSE yields $k = 10$ and $\alpha = 0$ as optimal hyperparameters. However, they lead to $F_{\boldsymbol{R}} > 0.25$, which indicates that many user-item pairs in off-blocks are considered similar. This is unreasonable for both the synthetic data generation process and real-world datasets. Hyperparameter tuning is a multi-objective problem, and in practice, the right balance between RMSE and folding is application-specific.

4.2 Assessing Folding on Real Data

The MovieLens 10M dataset contains 10M ratings for 10,681 movies by 71,567 users, yielding a sparsity rate of 1.3% [7]. We center the observed ratings $\boldsymbol{A}$ by subtracting the global average from all ratings and set the unobserved entries a_{ij} to 0. We randomly select 10% of all observed entries to be held out from training for testing. The optimization problem in Equation (1) is then solved using the training data and WALS with regularization $\lambda = 10$ while varying the embedding dimension k and unobserved weight α.

Computing the folding metric requires a definition of $\boldsymbol{R}$, the relatedness between all users and all items. Following the proposition in Section 3, we approximate $\mathbf{1}^{\Omega}$ using a rank l factorization $\boldsymbol{X}^{\top}\boldsymbol{Y}$, with $\boldsymbol{X} \in \mathbb{R}^{l\times m}$ and $\boldsymbol{Y} \in \mathbb{R}^{l\times n}$, using WALS with a high weight of

$\alpha = 1$ for unobserved ratings. Using this factorization, we define relatedness $\boldsymbol{R}^{\text{CF}}$ as

$$r_{ij}^{\text{CF}} = \max(0, \text{cosine}(\boldsymbol{x}_i, \boldsymbol{y}_j)) . \tag{3}$$

In our experiments, the $\boldsymbol{R}^{\text{CF}}$ matrices produced by setting l to 15, 30 and 35 lead to identical conclusions. All following results are obtained using $l = 30$.

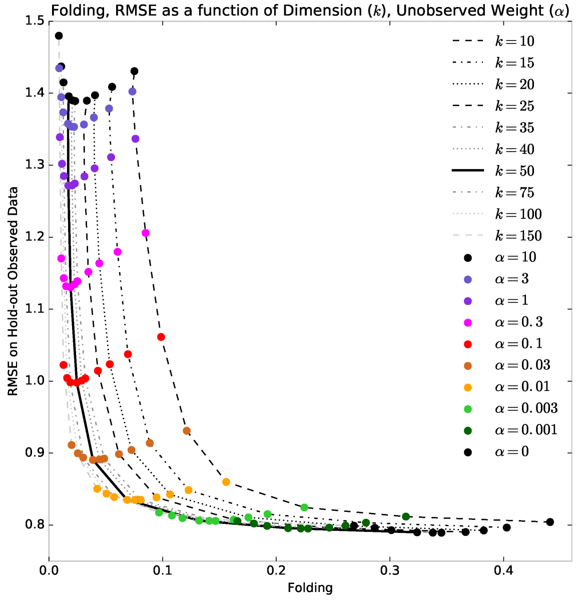

Figure 4: RMSE vs Folding on the MovieLens dataset.

Figure 4 shows the relationship between folding and RMSE as functions of k and α. RMSE is evaluated on hold-out data. We observe a negative correlation between RMSE and folding, which express two opposing goals. Consistent with the synthetic data, for any fixed dimension k, decreasing α improves RMSE, which drops rapidly from 1.5 to 0.9 with little increase in folding. However, once α is below 0.03, minor improvements in RMSE come at a heavy cost on folding. This suggests a compromise that would be overlooked when only RMSE is optimized. For any fixed α, increasing the dimension k reduces folding, as the embeddings gain more degrees of freedom. For left to right, the curves appear in order of decreasing k. Points with the same α and different k form parabolic curves in this order because both overfitting in the high dimensional regime and underfitting in the low dimensional regime lead to generalization error. The local minimum is consistently achieved at $k = 50$ (bold) for different values of α, which suggests that 50 is the closest to the true rank among our choices of embedding dimensions.

The best RMSE in our results is comparable to the state of the art on the same dataset (0.771 reported by [31]). However, all RMSE values ≤ 0.8 are obtained from low α (missing data is largely ignored) and associated with high folding measure ≥ 0.2. This suggests that the best-RMSE models have high propensity to make spurious recommendations like the ones in the first set of movies in Table 1.

From an information theoretic standpoint, k presents an upper bound on the amount of information that can be encoded in the embedding space. The model can choose to devote the information content to recovering the observed entries, avoiding overprediction on unrelated pairs, or a combination of both. Recovering observations and avoiding overpredictions are at odds with each other under limited model complexity. Suppose three distinct user-item groups exist in a dataset but the model is only capable of encoding two. The model can encode the two largest groups and omit the smallest, or it can merge two groups together so that observations in all three groups are recovered. In the former, the model misses out on all observation in the smallest group but does not fold; in the latter, the model overpredicts on unrelated user-item pairs that separate the two merged groups, leading to folding. This is the root cause behind the negative correlation observed between RMSE and folding.

4.3 Robustness of the Folding Metric

As mentioned in Section 4.2, comparing the folding metric obtained from different dimensions l of the approximate factorization of $\boldsymbol{1}^{\Omega}$ leads to similar results. To further demonstrate the robustness of the folding metric to estimations of $\boldsymbol{R}$, we compare the folding metric computed using $\boldsymbol{R}$ derived from a factorization-based estimation (using $l = 30$) with using a metadata-based approach.

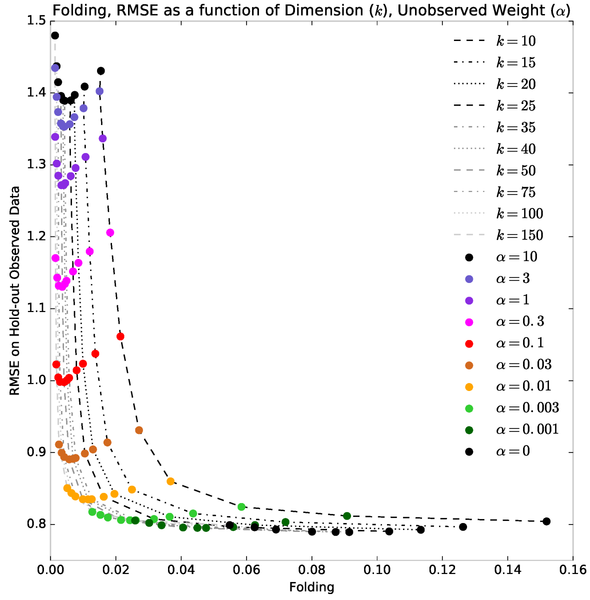

Figure 5: RMSE vs Folding on the MovieLens dataset, using genre embeddings for relatedness.

The MovieLens dataset [7] contains genre metadata, which can also form a basis for user-item relatedness. Each movie belongs to one or more of 18 genres. A simple model maps each movie j to an 18-dimensional vector, where y_{jd} is 1 if movie j belongs to genre d, and 0 otherwise. Each user is mapped onto the same space where x_{id} is the fraction of movies of genre d rated by user i. The relatedness $\boldsymbol{R}^{\text{genre}}$ is then defined by applying Equation (3) to the genre vectors.

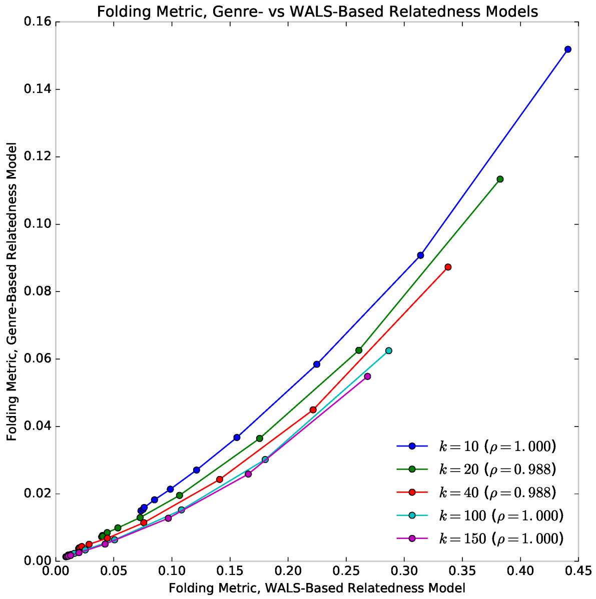

Figure 6: Folding metrics based on different estimates of R are strongly correlated.

Figure 5 shows the resulting parametric curves in the folding-RMSE space. The behavior is similar to the relationship between RMSE and folding in the case of the factorization-based estimation. In fact, the two choices for $\boldsymbol{R}$ lead to folding measures that are strongly monotonic, as shown in Figure 6. Each point in the plot represents a WALS model trained with a specific k and α. The set of models shown in Figure 6 is identical to the set in Figure 4. Points with the same k are connected to demonstrate the monotonicity between the folding measures obtained using the two choices for $\boldsymbol{R}$, evidenced by Spearman's $\rho \approx 1$.

5 RELATED WORK

The success of matrix factorization techniques in the Netflix Prize competition [17] popularized their use in modern recommender systems. One major strength of such approaches over their predecessors is the ability to incorporate implicit feedback, in contrast to explicit user-item ratings. Effectively augmenting recommender systems with implicit feedback requires understanding the nature of missing entries in the user-item matrix.

[19, 21] apply the missing data theory to address the issue of missing observations, and [21] establishes through a user study that data is missing not at random (MNAR), as users report to deliberately avoid rating certain groups of items, in hopes of improving their recommendations. [5, 8, 11, 12, 18, 20, 24, 25, 27, 29] investigate methods to distinguish missing data due to avoidance vs. lack of exposure. [11, 18, 27] propose to capture the missing data mechanism through a dedicated model that determines implicit negatives used for training the recommender system. [8] studies the low-rank nature of the missing data model and provides performance bounds derived from the low-rank assumption. [5, 12, 24] use heuristic approaches to weight the missing entries such that the matrix-factorization optimization objective is influenced by mispredictions on missing data, effectively treating missing data as weak implicit negative feedback. On the other hand, [29] provides alternative objectives designed to counter the bias introduced by MNAR data. Through comparative evaluations, authors in [20] conclude that MNAR algorithms outperform algorithms that assume data is missing at random (MAR). The proposed folding metric uses a low-rank similarity model that reflects the missing data mechanism to identify models that are prone to the pitfalls of MNAR data.

Effective offline evaluation, a challenge in recommendation model learning, has generated a great deal of research interest in recent years. Choosing the wrong evaluation metric can lead to suboptimal models for the recommendation task at hand [6]. The matter is complicated by the MNAR nature of data, which calls for evaluation frameworks that can differentiate different types of missing data. [6, 10, 28] provide surveys of existing metrics and their effectiveness on different recommendation tasks. [3] shows that for the top-N task, where the model predicts the N items with which a user is most likely to interact, algorithms that optimize for RMSE perform significantly worse than PureSVD (SVD with missing entries imputed as 0s). This result motivates our choices for ground truth similarity discussed earlier. [26] shows that measuring model performance on missing entries helps avoid bias introduced when considering only observed data. Our work complements existing metrics by measuring the severity of mispredictions on implicit unobserved data.

[15, 16, 22] study recommender system quality from a user-centric perspective. User satisfaction is often captured by a serendipity metric, complementary to the folding metric that quantifies bad user experience. [1, 4, 14] explore ways by which serendipity is measured and its inverse relationship to other metrics. [9] advocates for systems that provide explanations for its recommendations to improve user experience. The folding metric helps explain, to the system developers, potential causes of negative user experience.

6 CONCLUSION AND FUTURE WORK

Real-world recommender systems are often constrained by partially observed user-item interactions yet must generalize recommendations to unobserved user-item settings. We propose the folding metric to quantify the likelihood of producing incongruous recommendations while performing such generalizations. The key to this metric is the fact that not all generalizations are created equal. We have studied the metric in the low-rank matrix factorization context with WALS solvers, where unrelated items can be in close proximity in the embedding space due to lack of observations that explicitly force them apart. Our experiments with the MovieLens dataset [7] show that RMSE is negatively correlated with the folding metric, revealing that the best models trade off incremental RMSE gains for increasing folding costs. Experiments also affirm that the folding metric is robust to approaches used to estimate user-item relatedness, yielding similar results when using genre metadata as a proxy for relatedness vs. low-rank factorization. In future work, we aim to study the folding metric in the context of SGD solvers and develop negative sampling strategies that can effectively address folding in those systems.

REFERENCES

[1] Panagiotis Adamopoulos and Alexander Tuzhilin. 2015. On unexpectedness in recommender systems: Or how to better expect the unexpected. *ACM Transactions on Intelligent Systems and Technology* (2015).

[2] Konstantina Christakopoulou and Arindam Banerjee. 2015. Collaborative Ranking with a Push at the Top. *International World Wide Web Conference (WWW)* (2015).

[3] Paolo Cremonesi, Yehuda Koren, and Roberto Turrin. 2010. Performance of recommender algorithms on top-n recommendation tasks. In *RecSys*. ACM.

[4] Mouzhi Ge, Carla Delgado-Battenfeld, and Dietmar Jannach. 2010. Beyond accuracy: evaluating recommender systems by coverage and serendipity. In *RecSys*. ACM.

[5] Quanquan Gu, Jie Zhou, and Chris Ding. 2010. Collaborative filtering: Weighted nonnegative matrix factorization incorporating user and item graphs. In *International Conference on Data Mining*. SIAM.

[6] Asela Gunawardana and Guy Shani. 2009. A survey of accuracy evaluation metrics of recommendation tasks. *Journal of Machine Learning Research* (2009).

[7] F. Maxwell Harper and Joseph A. Konstan. 2015. The MovieLens Datasets: History and Context. *ACM Transactions on Interactive Intelligent Systems* (2015).

[8] Elad Hazan, Roi Livni, and Yishay Mansour. 2015. Classification with Low Rank and Missing Data. In *ICML*.

[9] Jonathan L Herlocker, Joseph A Konstan, and John Riedl. 2000. Explaining collaborative filtering recommendations. In *CSCW*. ACM.

[10] Jonathan L Herlocker, Joseph A Konstan, Loren G Terveen, and John T Riedl. 2004. Evaluating collaborative filtering recommender systems. *ACM Transactions on Information Systems (TOIS)* (2004).

[11] José Miguel Hernández-Lobato, Neil Houlsby, and Zoubin Ghahramani. 2014. Probabilistic Matrix Factorization with Non-random Missing Data. In *ICML*.

[12] Yifan Hu, Yehuda Koren, and Chris Volinsky. 2008. Collaborative filtering for implicit feedback datasets. In *International Conference on Data Mining (ICDM)*. IEEE.

[13] Sébastien Jean, Kyunghyun Cho, Roland Memisevic, and Yoshua Bengio. 2015. On Using Very Large Target Vocabulary for Neural Machine Translation. *Association for Computational Linguistics* (2015).

[14] Noriaki Kawamae. 2010. Serendipitous recommendations via innovators. In *SIGIR*. ACM.

[15] Bart P Knijnenburg, Martijn C Willemsen, Zeno Gantner, Hakan Soncu, and Chris Newell. 2012. Explaining the user experience of recommender systems. *User Modeling and User-Adapted Interaction* (2012).

[16] Joseph A Konstan and John Riedl. 2012. Recommender systems: from algorithms to user experience. *User Modeling and User-Adapted Interaction* (2012).

[17] Yehuda Koren, Robert Bell, and Chris Volinsky. 2009. Matrix factorization techniques for recommender systems. *IEEE Computer* (2009).

[18] Dawen Liang, Laurent Charlin, James McInerney, and David M Blei. 2016. Modeling user exposure in recommendation. In *International Conference on World Wide Web (WWW)*.

[19] Roderick JA Little and Donald B Rubin. 2014. *Statistical analysis with missing data*. John Wiley & Sons.

[20] Benjamin M Marlin and Richard S Zemel. 2009. Collaborative prediction and ranking with non-random missing data. In *RecSys*. ACM.

[21] Benjamin M. Marlin, Richard S. Zemel, Sam Roweis, and Malcolm Slaney. 2007. Collaborative Filtering and the Missing at Random Assumption. In *Uncertainty in Artificial Intelligence (UAI)*. AUAI Press.

[22] Sean M McNee, John Riedl, and Joseph A Konstan. 2006. Making recommendations better: an analytic model for human-recommender interaction. In *CHI extended abstracts on Human factors in computing systems*. ACM.

[23] Andriy Mnih and Koray Kavukcuoglu. 2013. Learning word embeddings efficiently with noise-contrastive estimation. In *NIPS*.

[24] Rong Pan and Martin Scholz. 2009. Mind the Gaps: Weighting the Unknown in Large-scale One-class Collaborative Filtering. In *KDD*. ACM.

[25] Rong Pan, Yunhong Zhou, Bin Cao, Nathan N. Liu, Rajan Lukose, Martin Scholz, and Qiang Yang. 2008. One-Class Collaborative Filtering. In *ICDM*. IEEE Computer Society.

[26] Bruno Pradel, Nicolas Usunier, and Patrick Gallinari. 2012. Ranking with non-random missing ratings: influence of popularity and positivity on evaluation metrics. In *RecSys*. ACM.

[27] Tobias Schnabel, Adith Swaminathan, Ashudeep Singh, Navin Chandak, and Thorsten Joachims. 2016. Recommendations as treatments: Debiasing learning and evaluation. *arXiv preprint arXiv:1602.05352* (2016).

[28] Guy Shani and Asela Gunawardana. 2011. Evaluating recommendation systems. In *Recommender systems handbook*. Springer, 257–297.

[29] Harald Steck. 2010. Training and testing of recommender systems on data missing not at random. In *KDD*. ACM.

[30] Hsiang-Fu Yu, Mikhail Bilenko, and Chih-Jen Lin. 2017. Selection of Negative Samples for One-class Matrix Factorization. In *SDM*. SIAM.

[31] Yin Zheng, Bangsheng Tang, Wenkui Ding, and Hanning Zhou. 2016. A Neural Autoregressive Approach to Collaborative Filtering. In *ICML*.