Ratings Re-specification for Rank Ordered Recommendations

Oluwasanmi Koyejo
Electrical Engineering Dept.
University of Texas at Austin

Sreangsu Acharyya
Electrical Engineering Dept.
University of Texas at Austin

Joydeep Ghosh
Electrical Engineering Dept.
University of Texas at Austin

Abstract

This paper introduces ratings re-specification; a novel framework for rank ordered recommendations. Ratings re-specification can be applied to any regression based recommendation model to optimize the rank ordering. The ratings are re-specified by searching for a monotonic transformation that results in a better fit while preserving their user-wise ranked order. In this paper, ratings re-specification is combined with a matrix factorization regression model to exploit shared low dimensional structure. We show that the resulting model recovers a unique solution under mild conditions, and propose a simple and efficient optimization scheme that alternates between re-specifying the ratings subject to ordering constraints, and matrix factorization regression. The re-specification step is independent for each user, and can be parallelized. The ranking performance of proposed approach is evaluated on benchmark movie recommendation datasets and results in superior ranking performance compared to recommender system algorithms specifically designed to optimize ranking metrics.

1 INTRODUCTION

Recommender systems are often trained to learn rating scores for each user-item pair. However, in most deployed systems, these learned scores are not shown to the user. Instead, the user is only shown a few of the top items as ordered by the learned scores [9]. This means that although regression metrics such as root mean square error (RMSE) and mean absolute error (MAE) are easier to optimize, ranking metrics such as normalized discounted cumulative gain (NDCG) [12] and expected reciprocal return (ERR) [7] are a more accurate reflection of recommender systems. Thus, the focus of research in the recommender systems literature has begun to shift from algorithms and metrics for *regression* to others that learn and measure the *ranking* of items for each user.

This paper proposes ratings re-specification; a framework for learning the user-wise ranking of items inspired by the study of learning to rank (LETOR) in the the information retrieval literature [17]. There, initial approaches using point-wise ranking models were replaced by *pair-wise* models [10], and are now being superseded by *list-wise* ranking models [6, 2]. The list-wise approach learns a ranking model for the entire set of items and has gained prominence in the LETOR literature with strong theoretical guarantees and superior empirical performance [18].

Ratings re-specification can be applied to any regression based recommendation model to optimize the rank ordering. We show that ratings specification inherits useful statistical and optimization theoretic properties when the loss function for the regression models is a *Bregman divergence* [5], a family of divergences that includes such popular loss functions such as squared loss and the Kullback-Leibler (KL) divergence.

The main contributions of this paper are as follows:

- We propose ratings re-specification; a framework used to transform a regression based recommendation model to a user-wise ranking model.
- We study the combination of ratings re-specification with matrix factorization (RR-MF).
- We propose a simple optimization scheme that alternates between the re-specification step and the regression step. The re-specification step is independent for each user and can be solved in parallel.
- Ratings re-specification with matrix factorization is evaluated on benchmark movie recommendation datasets and compared to recommender system algorithms specifically designed to optimize ranking metrics.

An extended version of this manuscript is published in [16].

Notation: Vectors are denoted by bold lower case letters, matrices are capitalized. $\boldsymbol{x}^\top$ denotes the transpose of the vector $\boldsymbol{x}$, $||\boldsymbol{x}||$ denotes the L_2 norm. A vector $\boldsymbol{x}$ is defined to be in *descending order* if $x_i \geq x_j$ when $i > j$, the set of such vectors is denoted by $\mathcal{R}\!\downarrow$. Vector $\boldsymbol{x}$ is isotonic with $\boldsymbol{y}$ if $x_i \geq x_j$ implies $y_i \geq y_j$. The unit simplex is denoted by Δ and Δ_ϵ denotes the subset of an unit simplex such that each of its members are component-wise bounded away from 0 by ϵ. The positive orthant is denoted by $\mathcal{R}^d_+$ and $\mathcal{R}^d_\epsilon$ denotes its subset such that each of its members are component-wise bounded away from 0 by ϵ.

1.1 RELATED WORK

Models for user-wise ranking have been studied by several researchers in the recommender systems literature. Point-wise models applied to collaborative filtering predict user ratings using regression or classification methods, the final ranking is defined by the ordering of the regression scores. Cremonesi et al. [9] showed that methods trained to optimize regression metrics may not be effective for top-k recommendation. Steck et al. [21] proposed modifications of matrix factorization methods motivated by top-k ranking performance. A related class of point-wise ordinal regression models have also been proposed. These models are optimized so that user ratings are correctly placed in ordered bins corresponding to the different rating levels. For instance, maximum margin matrix factorization (MMMF) [20] jointly optimized the matrix factorization and the user rating bins using the Hinge loss, and Ordrec [15] combined logistic ordinal regression loss with matrix factorization.

Ranking performance may be improved further by using a pair-wise approach. Balakrishnan et al. [4] proposed a pairwise classification approach for collaborative ranking. The authors showed effective ranking performance as measured using the NDCG metric. Pair-wise ranking results in a model with computational cost that is quadratic in the number of items to be ranked. This is a significant increase in computational cost and may be prohibitive in large scale recommender systems with millions of ratings Another concern is the fact that pair-wise orders are not necessarily transitive. Hence, orders over pairs of items may not directly translate into an ordered list. For example, given three items $\{a, b, c\}$, there is no consistent ordered list if the pair-wise relations are given by $\{a > b\}, \{b > c\}, \{c > a\}$.

List-wise approaches avoid the computational and consistency hazards of pair-wise methods. In addition, list-wise methods often have stronger statistical and optimization theoretic guarantees [18, 2], and superior empirical performance. List-wise models applied to recommender systems are known as user-wise ranking models. CoFi^Rank [22], a popular approach for user-wise recommender systems, trains a matrix factorization model to optimize a bound of the NDCG metric. Shi et al. [19] adapted the list-wise ranking algorithm proposed in [6] to recommender systems by replacing the underlying linear regression model with a matrix factorization regression model.

1.2 BACKGROUND ON BREGMAN DIVERGENCES

Let $\boldsymbol{\phi} : \Theta \mapsto \mathbb{R},\ \Theta = \operatorname{dom}\boldsymbol{\phi} \subseteq \mathbb{R}^d$ be a strictly convex, closed function, differentiable on $\operatorname{int}\Theta$. The corresponding Bregman divergence $D_\phi(\cdot\|\cdot) : \operatorname{dom}(\boldsymbol{\phi}) \times \operatorname{int}(\operatorname{dom}(\boldsymbol{\phi})) \mapsto \mathbb{R}_+$ is defined as $D_\phi(\boldsymbol{x}\|\boldsymbol{y}) \triangleq \boldsymbol{\phi}(\boldsymbol{x}) - \boldsymbol{\phi}(\boldsymbol{y}) - \langle \boldsymbol{x} - \boldsymbol{y}, \nabla\boldsymbol{\phi}(\boldsymbol{y})\rangle$. In this paper we only consider functions of the form $\boldsymbol{\phi}(\cdot) : \mathbb{R}^n \ni \boldsymbol{x} \mapsto \sum_i \phi(x_i)$ that are sums of *identical* scalar convex functions applied to each component. We refer to this class as *identically separable* (**IS**). This class has properties particularly suited to ranking. Squared loss and Kullback-Leibler divergence (KL) are in this family.

2 LEARNING TO RECOMMEND

Let $\mathcal{U}$ and $\mathcal{V}$ denote the set of users and items respectively. Let $\mathcal{V}_i$ denote the set of items rated by user i and $|\mathcal{V}_i|$ its cardinality. We denote the true ratings by Y_{ij} and predicted ratings by $\hat{Y}_{ij}$. Let w_{ij} be a binary variable that denotes whether user i has rated item j ($w_{ij} = 1$) or not ($w_{ij} = 0$). The item recommendation task consists of learning the user's taste from the training ratings. In matrix factorization (MF) [14], the predictions take the form:

$$\hat{Y}_{ij} = \langle \boldsymbol{u}_i, \boldsymbol{v}_j \rangle, \tag{1}$$

where $\boldsymbol{u}_i \in \mathbb{R}^d$ is called a user factor for user i and $\boldsymbol{v}_j \in \mathbb{R}^d$ is the item factor for item j.

The factors are learned by minimizing the squared Euclidean distance between the actual ratings and the predicted ratings as indicated in equation (2):

$$\min_{\boldsymbol{u}_i, \boldsymbol{v}_j} \sum_{i \in \mathcal{U}, j \in \mathcal{V}} \frac{1}{2} w_{ij} (Y_{ij} - \hat{Y}_{ij})^2. \tag{2}$$

In this context the cost function (2) is unnecessarily strict. It is only required that the predictions $\hat{Y}_{ij}$ order the items in the same way that Y_{ij} does.

These drawbacks are now well recognized and have inspired several ranking based approaches to the the recommendation problem. The newer approaches replace (2) by other cost functions that depend not on the predicted scores themselves, but on the order induced by them. Examples of such cost functions include the normalized cumulative discounted gain (NDCG) [12], expected reciprocal return (ERR) [7] and mean absolute precision (MAP) [3]. Their use however introduce a significant difficulty. The domain of these cost functions is the space of permutations of the list of items. This is not only a discrete (combinatorial) space, but also one whose size grows exponentially with the number of items. This makes training with respect to these cost functions frightfully difficult.

In this paper we adapt a list-wise learning to rank (LETOR) algorithm called monotone retargeting (MR), developed recently [2], to the recommendation task. MR introduces a new family of cost functions that have several desirable properties, both computational and statistical. It uses a cost function that is truly a function of the order and not of the predicted values, therefore well suited for ranking. For appropriate choices, the cost function is jointly convex in its parameters, hence has a global minimum. MR allows efficient as well as highly parallelizable algorithms for training. The key observations that motivates the MR is that (i) the combinatorial problem of LETOR can be transformed into a problem of searching over the space of all monotonic transformations and (ii) it is possible to design an efficient, convergent technique to solve the challenging task of minimizing over this infinite function space without sacrificing generality.

2.1 RATINGS RE-SPECIFICATION WITH MATRIX FACTORIZATION

Given any loss function $D : \mathbb{R} \times \mathbb{R} \mapsto \mathbb{R}_+$, we may define a pointwise LETOR based MF problem by:

$$\min_{\boldsymbol{u}_i, \boldsymbol{v}_j, \Upsilon_i \in \mathcal{M}} \sum_i \sum_j w_{ij} D\big(Y_{ij}, \Upsilon_i \circ f(\langle \boldsymbol{u}_i, \boldsymbol{v}_j \rangle)\big),$$

where $f : \mathbb{R} \mapsto \mathbb{R}$ is some regression function and $\Upsilon_i : \mathbb{R} \mapsto \mathbb{R}$ is a monotonic increasing transformation and $\mathcal{M}$ is the class of all such transformations. As noted in [2] the optimization over the infinite space of functions $\mathcal{M}$ can be converted into one over finite dimensional vector spaces provided we have a finite characterization of the constraint set $\mathcal{R}{\downarrow}_i$. Let $Y_{i,*}$ represent the item ratings for user i arranged in (non-unique) sorted order. Further, let $V \in \mathbb{R}^{|\mathcal{V}| \times d}$ with $V(j) = \boldsymbol{v}_j$ represent the collected item factors, and $V_i \in \mathbb{R}^{|\mathcal{V}_i| \times d}$ represent the subset of item factors corresponding to the items rated by user i, sorted to match the ratings $Y_{i,*}$. Without loss of generality, the monotone transformation can be applied to the left hand side i.e. to the sorted ratings. The resulting cost function is:

$$\min_{\boldsymbol{u}_i, V, \boldsymbol{r}_i \in \mathcal{R}{\downarrow}_i} \sum_i D(\boldsymbol{r}, f(V_i \boldsymbol{u}_i)) \tag{3}$$
$$\text{s.t. } \mathcal{R}{\downarrow}_i = \{\boldsymbol{r} |^{\exists \boldsymbol{M} \in \mathcal{M}}_{\boldsymbol{M}(Y_{i,*}) = \boldsymbol{r}}\}.$$

where $f(\cdot)$ is applied element-wise to result of the matrix vector product $V_i \boldsymbol{u}_i$, and $\mathcal{R}{\downarrow}_i$ represents all vectors that are sorted in decreasing order. Hence $\mathcal{R}{\downarrow}_i$ hence includes vectors $\boldsymbol{r} \in \mathbb{R}^{|\mathcal{V}_i|}$ such that $r_{k+1} = r_k$ or some k. Since the vectors $\boldsymbol{r}$ are the targets given to the regressor to fit, it is more robust to enforce separation between the components, i.e. maintain $r_k \geq r_{k+1} + \epsilon$. We indicate the set of all such ϵ separated decreasing ordered sets by $\mathcal{R}_\epsilon{\downarrow}_i$.

The Set $\mathcal{R}_\epsilon{\downarrow}_i$: The convex composition $\boldsymbol{r} = \alpha \boldsymbol{r}_1 + (1 - \alpha)\boldsymbol{r}_2$ of two isotonic vectors $\boldsymbol{r}_1$ and $\boldsymbol{r}_2$ preserves isotonicity, as does the scaling $\alpha \boldsymbol{r}$ for any $\alpha \in \mathbb{R}_+$. Hence the set $\mathcal{R}_\epsilon{\downarrow}_i$ is a convex cone.

In addition, we shall make use of the set of all discrete probability distributions sorted in decreasing order but also satisfying $r_i \geq r_{i+1} + \frac{\epsilon}{i}$. i.e. $\mathcal{R}_\epsilon{\downarrow}_i \cap \Delta_i$ that we represent by $\Delta_\epsilon{\downarrow}$. Note that vectors in $\Delta_\epsilon{\downarrow}$ are not only sorted but have an increasing gap between consecutive components.

Lemma 1 *The set $\Delta_\epsilon{\downarrow}$ of all discrete probability distributions of dimension d that are in descending order is the image $T\boldsymbol{x}$ s.t. $\boldsymbol{x} \in \Delta_\epsilon$ where T is an upper triangular matrix generated from the vector $\boldsymbol{v}_\Delta = \{1, \frac{1}{2} \cdots \frac{1}{d}\}$ such that $T(i, :) = \{0\}^{i-1} \times \boldsymbol{v}_\Delta(i :)$*

The motivation for such a construction is that even if the regressor is inaccurate, if the inaccuracy is more or less uniform across the components, the top entries of the predictions will be less prone to order reversal than the bottom ones: a desirable feature for ranking. We note that MR [2] did not consider any of the ϵ separated ordered sets we have described in this section.

With appropriate choices of the distance like function $D(\cdot, \cdot)$ and the curve fitting function $f(\cdot)$ we can transform (3) into a bi-convex optimization problem over a product of convex sets. We choose $D(\cdot, \cdot)$ to be a Bregman divergence $D_\phi\big(\cdot \big|\big| \cdot\big)$, defined in Section 1.2, and $f(V_i \boldsymbol{u}_i)$ to be $(\nabla_\phi)^{-1}(V_i \boldsymbol{u}_i)$, leading to the formulation:

$$\min_{\boldsymbol{u}_i, V, \boldsymbol{r} \in \mathcal{R}_\epsilon{\downarrow}_i} \sum_i D_{\boldsymbol{\phi}}\big(\boldsymbol{r}_i \big|\big| (\nabla_\phi)^{-1} (V_i \boldsymbol{u}_i)\big). \tag{4}$$

Bregman divergences are the class of regression loss functions of choice, motivated by various statistical and optimization theoretic properties. Ravikumar et al., [18] showed that Bregman divergences are the *only family* of cost functions that are strongly consistent with the NDCG metric. In addition, we highlight the property of joint convexity. Let $\boldsymbol{z}_i = V_i \boldsymbol{u}_i$ represent the vector of predicted ratings of user i. Acharyya et al. [2], showed that (4) is jointly convex in $\{\boldsymbol{r}_i\}$ and $\{\boldsymbol{z}_i\}$ if and only if the Bregman divergence is squared loss.

We note that since we maintain explicit representation of the factor matrices U and V, the optimization problem is no longer convex with respect to these factors. However we can apply the results of [1] (Proposition 5) which show that if the cost function is strongly convex and the selected rank is sufficiently large, the local minima in terms of U and V are global in terms of the regression matrix $Z = UV^\top$. It follows that under mild conditions, RR-MF using squared loss recovers a unique solution. The cost function using other Bregman divergences can only provide local optimality guarantees.

3 FORMULATION AND ALGORITHM

We safeguard against overfitting by adding squared Frobenius norm regularization for the matrices U and V. Note that the cost function (4) is not invariant to scale. i.e. the cost can be reduced just by scaling its arguments down, without actually learning the task. To remedy this, we constrain $\boldsymbol{r}_i$s to lie in an appropriate closed convex set not only separated from the origin but also to a set of vectors whose adjacent components are separated from each other. For the latter we use the set $\Delta_{\epsilon\downarrow}$ as defined in Section 2.1. After these modifications we obtain the final formulation as:

$$\min_{\boldsymbol{u}_i, V, \boldsymbol{r}_i \in \Delta_{\epsilon\downarrow i}} \sum_i D_{\boldsymbol{\phi}}\big(\boldsymbol{r}_i \big|\big| (\nabla_\phi)^{-1} (V_i \boldsymbol{u}_i)\big) + \frac{\lambda_u}{2} \sum_i ||\boldsymbol{u}_i||^2 + \frac{\lambda_v}{2} \sum_j ||\boldsymbol{v}_j||^2 \quad (5)$$

In our formulation we assume that the true movie ratings are totally ordered, though the finer ordering between similar items is not visible to the ranking algorithm. Let $P_j = \{P_{jk}\}_{k=1}^{k_j}$ be a partition of the index set of $\mathcal{V}_j$, such that all items in P_{jk} have the same training score. (for example the set of all movies rated 5 by a particular user). The sets $\mathcal{V}_j$ effectively get partitioned further into $\{P_{jk}\}_{k=1}^{k_j}$. Though the ratings provide an order between movies from any two different sets P_{jk} and P_{jl}, the order within any set P_{jk} remains unknown. To retrieve the total order we introduce a block-diagonally restricted permutation matrix $\mathbb{P}_j$ that can permute indices in each P_{jk} independently. Since the items in P_{jk} are not equivalent they are available for re-ordering as long as that minimizes the cost (5). Using the properties of Bregman divergence, Acharyya et al. [2] showed that if $\boldsymbol{\phi}(\cdot)$ is IS, the solution is given by the sorted order. Thus update (6) can be accomplished by sorting. The combined algorithm for RR-MF is given in Fig. 1. We cycle through all three update steps until convergence.

$$\mathbb{P}_i^{t+1} = \underset{\pi}{\text{Argmin}}\, D_{\boldsymbol{\phi}}\big(T\boldsymbol{x}_i^t \big|\big| (\nabla_\phi)^{-1} (\pi V_i \boldsymbol{u}_i)\big) \quad \forall i \text{ parallel} \quad (6)$$

$$\boldsymbol{x}_i^{t+1} = \underset{\boldsymbol{x} \in \Delta_{\epsilon\downarrow}}{\text{Argmin}}\, D_{\boldsymbol{\phi}}\big(T\boldsymbol{x} \big|\big| (\nabla_\phi)^{-1} \left(\mathbb{P}_i^{t+1} V_i \boldsymbol{u}_i\right)\big) \quad \forall i \text{ parallel} \quad (7)$$

$$U, V = \underset{\{\boldsymbol{u}_i, V\}}{\text{Argmin}} \sum_i D_{\boldsymbol{\phi}}\big(T\boldsymbol{x}_i^{t+1} \big|\big| (\nabla_\phi)^{-1} \left(\mathbb{P}_i^{t+1} V \boldsymbol{u}\right)\big) + \frac{\lambda_u}{2} \sum_i ||\boldsymbol{u}_i||^2 + \frac{\lambda_v}{2} \sum_j ||\boldsymbol{v}_j||^2 \quad (8)$$

Figure 1: RR-MF Optimization Algorithm

4 EXPERIMENTS

We evaluated RR-MF on three publicly available recommender datasets:

- Movielens[1] is a movie recommendation website administered by GroupLens Research. GroupLens Research has made available three ratings datasets of varying sizes[2]. We used the movielens 1M and movielens 10M datasets. Ratings in movielens 1M take one of 5 values in the set $\{1.0, 2.0, \ldots, 5.0\}$. Ratings in movielens 10M take one of 10 values in the set $\{0.5, 1, 1.5, \ldots, 5.0\}$
- Flixster[3] is a website where users share film reviews and ratings. We used the flixster dataset provided by [11] with timestamps. Ratings in Flixster take one of 10 values in the set $\{0.5, 1, 1.5, \ldots, 5.0\}$.

Table 1: Data Sizes After Preprocessing

DATASET	# USERS	# ITEMS	# RATINGS
Movielens 1M	5,289	3,701	982,040
Movielens 10M	57,534	10,675	9,704,223
Flixster	30,277	48,146	7,580,563

Data preprocessing: First, we removed all users with less than 30 ratings. Each of the evaluated datasets contains the time-stamp of the user rating. For each user, we selected the last third of the ratings sorted by user-time as the test set, the middle third as the validation set, and any left over was selected as the training set. This ensures that each user has at least 10 ratings in the validation and test sets so we are able to compute the top-k performance metrics for at least 10 retrieved items per user. Details of the dataset sizes after preprocessing are provided in table Table 1.

We experimented with RR-MF using squared loss based on the optimization theoretic guarantees. RR-MF was implemented in Python/Numpy. We implemented (7) using the exponentiated gradient (EG) algorithm [13]. Cython was used to implement the parallel re-specification updates (7) and parallel sorting (6). We also evaluated the performance of CoFiRank -NDCG [22] and CoFiRank -ordinal [23] as baseline models using the C++ implementation provided by the authors [4].

The models are scored using the following metrics commonly used for evaluating ranking models [17]: Normalized discounted cumulative gain (NDCG$_{@k}$), precision (P$_{@k}$), expected reciprocal return (ERR) and mean average precision (MAP). For all models, we selected the

[1] movielens.umn.edu
[2] www.grouplens.org/node/73
[3] www.flixster.com
[4] available at http://www.cofirank.org

regularization parameter $\lambda = \lambda_u = \lambda_v$ from the set $10^{\{-2,-1.5,-1,\ldots,2\}}$. We plot NDCG and precision in Fig. 2. Further results with other ranking metrics were also computed including ERR (Table 2), MAP (Table 2)) and the NDCG of the full list (Table 4)). For the precision and MAP metrics, a movie was considered relevant if its rating was greater than 4. In all tables, "–" represents datasets where CoFi$^{\text{RANK}}$-NDCG did not finish after running for more than seven days.

Our experiments show that RR-MF improves ranking performance over CoFi$^{\text{RANK}}$-NDCG and CoFi$^{\text{RANK}}$-ordinal as measured by all the metrics that we computed. We note that CoFi$^{\text{RANK}}$-ordinal [23] has been shown to outperform several state of the art models including maximum margin matrix factorization [20] and Gaussian process ordinal regression [8]. The results were even more striking when we compared the NDCG performance of RR-MF to CoFi$^{\text{RANK}}$-NDCG, though the algorithm is specifically designed to optimize NDCG. Our results were qualitatively very similar for rank 10 and rank 20 models.

The code was executed on a 2.4GHz quad-core Intel Xeon processor. Timing on the larger movie datasets are shown in Table 5 and compared to CoFi$^{\text{RANK}}$. We found that RR-MF exhibited much better scaling behavior as the size data increased. We suspect that the large observed runtimes of CoFi$^{\text{RANK}}$-NDCG are due to the cost of the linear assignment problem that must be solved for each user at every iteration. The computational requirements of the linear assignment problem scale cubically with the number of ratings per user. RR-MF is able to avoid solving this linear assignment problem since for Bregman divergences, sorting recovers the optimal ordering.

Table 2: Expected Reciprocal Return (ERR)

	RR-MF	**CoFi$^{\text{RANK}}$-NDCG**	**CoFi$^{\text{RANK}}$-ORD.**
Movielens 1M			
Rank 10	**0.819**	0.726	0.774
Rank 20	**0.816**	0.728	0.756
Movielens 10M			
Rank 10	**0.780**	0.687	0.763
Rank 20	**0.781**	–	0.753
Flixster			
Rank 10	**0.771**	–	0.743
Rank 20	**0.771**	–	0.737

5 CONCLUSION AND FUTURE WORK

In this paper, we proposed ratings re-specification with matrix factorization (RR-MF), a novel approach for learning the user-wise ranking of items for recommender systems. Ratings re-specification improves the ranking performance by searching for a monotonic transformation of the ratings that that is a results in a better fit while preserving the ranked order of the ratings. RR-MF was compared to the ranking and ordinal regression variants of CoFi$^{\text{RANK}}$ and evaluated on benchmark movie recommendation datasets. Our results show superior ranking performance compared to CoFi$^{\text{RANK}}$-NDCG, though CoFi$^{\text{RANK}}$-NDCG is specifically designed to optimize NDCG. We plan to explore the use of other Bregman divergences in future work. Although the unique solution is only guaranteed for squared loss, the performance results presented in [2] provide some motivation for applying other divergences. We also plan to explore the use of side information to improve the user-wise ranking performance.

Table 3: Mean Absolute Precision (MAP)

	RR-MF	**CoFi$^{\text{RANK}}$-NDCG**	**CoFi$^{\text{RANK}}$-ORD.**
Movielens 1M			
Rank 10	**0.485**	0.358	0.435
Rank 20	**0.482**	0.360	0.408
Movielens 10M			
Rank 10	**0.472**	0.353	0.453
Rank 20	**0.470**	–	0.439
Flixster			
Rank 10	**0.513**	–	0.488
Rank 20	**0.509**	–	0.480

Table 4: Normalized Cumulative Discounted Gain (NDCG)

	RR-MF	**CoFi$^{\text{RANK}}$-NDCG**	**CoFi$^{\text{RANK}}$-ORD.**
Movielens 1M			
Rank 10	**0.896**	0.846	0.877
Rank 20	**0.895**	0.847	0.867
Movielens 10M			
Rank 10	**0.892**	0.843	0.883
Rank 20	**0.891**	–	0.878
Flixster			
Rank 10	**0.888**	–	0.877
Rank 20	**0.887**	–	0.874

References

[1] J. Abernethy, F. Bach, T. Evgeniou, and J.-P. Vert. A new approach to collaborative filtering: Operator estimation with spectral regularization. *JMLR*, 10, 2009.

[2] S. Acharyya, O. Koyejo, and J. Ghosh. Learning to rank with Bregman divergences and monotone retargeting. In *UAI*, 2012.

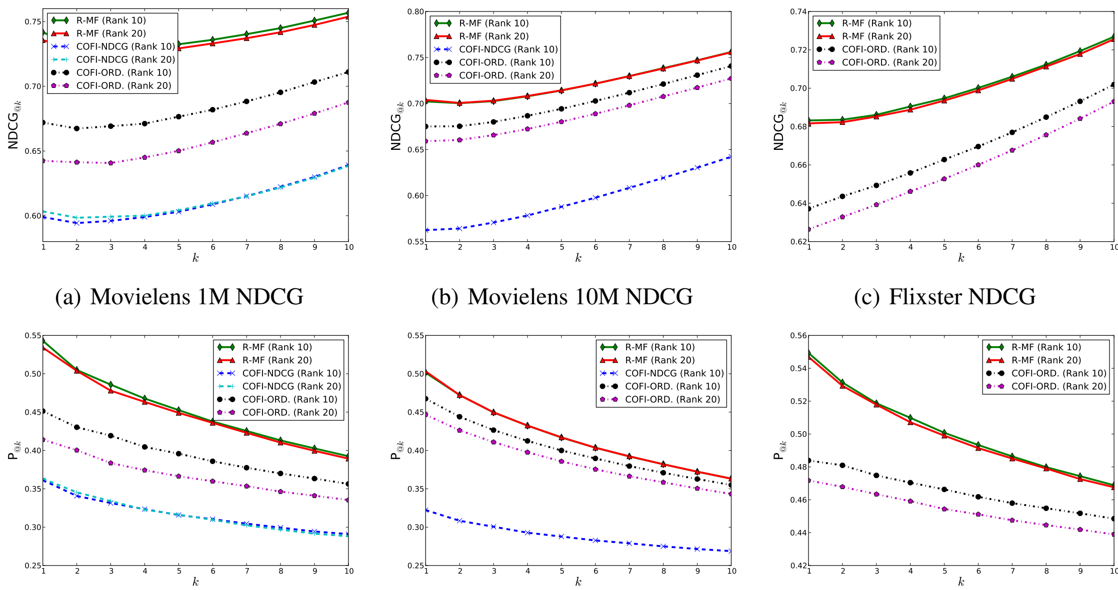
[3] R. A. Baeza-Yates and B. Ribeiro-Neto. *Modern Information Retrieval*. Addison-Wesley Longman Publishing Co., Inc., 1999.

[4] S. Balakrishnan and S. Chopra. Collaborative ranking. In *WSDM*, 2012.

[5] A. Banerjee, S. Merugu, I. Dhillon, and J. Ghosh. Clustering with Bregman divergences. *JMLR*, 6, 2005.

[6] Z. Cao, T. Qin, T.-Y. Liu, M.-F. Tsai, and H. Li. Learning to rank: from pairwise approach to listwise approach. In *ICML*, 2007.

[7] O. Chapelle, D. Metlzer, Y. Zhang, and P. Grinspan. Expected reciprocal rank for graded relevance. In *CIKM*, 2009.



(a) Movielens 1M NDCG (b) Movielens 10M NDCG (c) Flixster NDCG

(d) Movielens 1M Precision (e) Movielens 10M Precision (f) Flixster Precision

Figure 2: $\text{NDCG}_{@k}$ and $\text{P}_{@k}$ at $@k = 1, 2, \ldots, 10$. CoFi$^{\text{Rank}}$ -ordinal [23] has been shown to outperform to several state of the art models including maximum margin matrix factorization [20] and Gaussian process ordinal regression [8].

Table 5: Average training times (mins) on the larger recommender system datasets. "–" represents datasets where CoFi$^{\text{Rank}}$ -NDCG did not finish after running for more than seven days.

	RR-MF	CoFi$^{\text{Rank}}$ -NDCG	CoFi$^{\text{Rank}}$ -ORD.
Movielens 1M			
Rank 10	121.025	**329.889**	41.641
Rank 20	259.051	**340.569**	30.276
Movielens 10M			
Rank 10	396.900	**2912.179**	1391.249
Rank 20	663.889	**7609.683**	820.579
Flixster			
Rank 10	1657.314	–	2459.418
Rank 20	2203.207	–	897.989

[8] W. Chu and Z. Ghahramani. Gaussian processes for ordinal regression. *JMLR*, 6, 2004.

[9] P. Cremonesi, Y. Koren, and R. Turrin. Performance of recommender algorithms on top-n recommendation tasks. In *Recsys*, 2010.

[10] Y. Freund, R. Iyer, R. E. Schapire, and Y. Singer. An efficient boosting algorithm for combining preferences. *J. Mach. Learn. Res.*, 4:933–969, 2003.

[11] M. Jamali and M. Ester. A matrix factorization technique with trust propagation for recommendation in social networks. In *RecSys*, pages 135–142, 2010.

[12] K. Järvelin and J. Kekäläinen. IR evaluation methods for retrieving highly relevant documents. In *Proceedings of the 23rd annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '00, pages 41–48, New York, NY, USA, 2000. ACM.

[13] J. Kivinen and M. K. Warmuth. Exponentiated gradient versus gradient descent for linear predictors. *Information and Computation*, 132, 1995.

[14] Y. Koren, R. Bell, and C. Volinsky. Matrix factorization techniques for recommender systems. *Computer*, 42:30–37, 2009.

[15] Y. Koren and J. Sill. Ordrec: an ordinal model for predicting personalized item rating distributions. In *RecSys*, 2011.

[16] O. Koyejo, S. Acharyya, and J. Ghosh. Retargeted matrix factorization. In *Recsys*. ACM, 2013.

[17] T. Liu. *Learning to Rank for Information Retrieval*. Information retrieval. Springer, 2011.

[18] P. Ravikumar, A. Tewari, and E. Yang. On NDCG consistency of listwise ranking methods. In *AISTATS*, 2011.

[19] Y. Shi, M. Larson, and A. Hanjalic. List-wise learning to rank with matrix factorization for collaborative filtering. In *Recsys*, 2010.

[20] N. Srebro, J. D. M. Rennie, and T. S. Jaakola. Maximum-margin matrix factorization. In *NIPS*, 2005.

[21] H. Steck. Training and testing of recommender systems on data missing not at random. In *KDD*, 2010.

[22] M. Weimer, A. Karatzoglou, Q. V. Le, and A. Smola. Cofi-rank, maximum margin matrix factorization for collaborative ranking. In *NIPS*, 2007.

[23] M. Weimer, A. Karatzoglou, and A. J. Smola. Improving maximum margin matrix factorization. *Machine Learning*, 72:263–276, 2008.