Convex AUC Optimization for Top-N Recommendation with Implicit Feedback

Fabio Aiolli
University of Padova, Italy
aiolli@math.unipd.it

ABSTRACT

In this paper, an effective collaborative filtering algorithm for top-N item recommendation with implicit feedback is proposed. The task of top-N item recommendation is to predict a ranking of items (movies, books, songs, or products in general) that can be of interest for a user based on earlier preferences of the user. We focus on implicit feedback where preferences are given in the form of binary events/ratings. Differently from state-of-the-art methods, the method proposed is designed to optimize the AUC directly within a margin maximization paradigm. Specifically, this turns out in a simple constrained quadratic optimization problem, one for each user. Experiments performed on several benchmarks show that our method significantly outperforms state-of-the-art matrix factorization methods in terms of AUC of the obtained predictions.

Categories and Subject Descriptors

H.4 [**Information Systems Applications**]: Miscellaneous; H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval

Keywords

Collaborative Filtering; Top-N Recommendation; Implicit Feedback; AUC optimization

1. INTRODUCTION

Collaborative filtering aims at improving customer experience in e-commerce applications. These techniques exploit the history of user interactions to improve future recommendations to users. Different types of interactions can be stored and used. For example, a user can be asked a vote, a rate, or a degree of satisfaction for some of the items available. The system can also keep the history of user purchases autonomously, browsing activity of the users, etc. The former case is commonly called *explicit feedback* while the latter is usually referred to as *implicit feedback*.

In a collaborative recommender system we have a set $\mathbb{U}$ of n users, a set $\mathbb{I}$ of m items, and a user-item rating matrix $\mathbf{R} = \{r_{ui}\} \in \mathbb{R}^{n\times m}$ representing, for any user and item, how much user u likes item i. In this paper, we assume implicit feedback and binary ratings, that is $r_{ui} \in \{0, 1\}$. A value $r_{ui} = 1$ means that the user u rated the item i, while a value $r_{ui} = 0$ means not availability of information about whether user u likes item i.

There are two principal recognized approaches to collaborative filtering [1]. On one hand, model-based collaborative filtering, a.k.a. matrix factorization (MF) aims at constructing a model of the matrix $\mathbf{R}$. There are several proposed methods of this type, including Bayesian models, clustering models, classification, and regression models. On the other side, there are memory-based (MB) collaborative filtering methods where similarities among users (or items) are computed on the basis of the training ratings and the prediction is made on the basis of these similarities.

As shown in this paper, the two approaches above can be seen in the same optimization framework where a matrix factorization has to be found. Specifically, in the memory based approaches, like user-based and item-based nearest neighbors, particular factorizations are fixed a-priori and no actual train needs to be performed.

Despite MF is recognized to be the state-of-the-art technique in collaborative filtering, typically much better than MB techniques, MF has some drawbacks that made it impractical and unsuccessful sometimes. First of all, training the corresponding model is computationally onerous and this fact is crucial when the size of the matrix $\mathbf{R}$ is very large. Second, since MF is typically modelled as a regression problem, then it seems unsuitable for implicit feedback tasks and some modifications to the original technique are needed [4]. Third, MF techniques typically solve the associated non convex minimization problem by using gradient descent algorithms which do not guarantee the convergence to a global minimum and, it is well known, the rate of convergence is quite low when near to local minima. Finally, the high number of parameters to optimize, the sparsity of the rating matrix, and no apriori knowledge available can possibly lead to overfitting and strong validation and/or early-stopping of the stochastic gradient descent is often necessary.

In this paper we propose a solution lying midway between the two extreme approaches of MB and MF. In particular, we cast the problem into a instance ranking one and we propose a convex quadratic optimization framework for the optimization of AUC.

2. TOP-N RECOMMENDATION

In the top-N recommendation setting, for each user u, there are available a subset $\mathbb{I}(u) \subset \mathbb{I}$ of items that have been rated by the user and the task is to predict a ranking of the items not yet rated such

RecSys'14, October 6–10, 2014, Foster City, Silicon Valley, CA, USA.
Copyright 2014 ACM 978-1-4503-2668-1/14/10 ...$15.00.
http://dx.doi.org/10.1145/2645710.2645770.

that most of another set $\bar{\mathbb{I}}(u) \subseteq \mathbb{I} - \mathbb{I}(u)$ of items are in the first N positions of the ranking. In this paper, we indicate as $\mathbb{U}(i) = \{u \in \mathbb{U} | i \in \mathbb{I}(u)\} \subset \mathbb{U}$ the subset of users that have rated the item i in the initial set and $\bar{\mathbb{U}}(i) = \{u \in \bar{\mathbb{U}} | i \in \bar{\mathbb{I}}(u)\} \subseteq \mathbb{U} - \mathbb{U}(i)$ the subset of users that have rated the item i in the test set. By construction, the sets are chosen such that $\mathbb{I}(u) \cap \bar{\mathbb{I}}(u) = \phi$. We also define $m_u^+ = |\mathbb{I}(u)|$ and $m_u^- = m - m_u^+$.

In the following we indicate by $\mathbf{R} = \{r_{ui}\} \in \mathbb{R}^{n \times m}$ the initial rating matrix such that $r_{ui} = 1$ whenever $i \in \mathbb{I}(u)$ and $r_{ui} = 0$ otherwise, and by $\bar{\mathbf{R}} = \{\bar{r}_{ui}\} \in \mathbb{R}^{n \times m}$ the test rating matrix such that $\bar{r}_{ui} = 1$ whenever $i \in \bar{\mathbb{I}}(u)$ and $\bar{r}_{ui} = 0$ otherwise.

3. COLLABORATIVE FILTERING

In this section, we consider a general framework where many different approaches can be cast. Let the matrix $\mathbf{W} \in \mathbb{R}^{n \times k}$ be the embedding of users in a factor space of dimension k with users representations $\mathbf{w}_u \in \mathbb{R}^k$ as rows. Similarly, let the matrix $\mathbf{X} \in \mathbb{R}^{k \times m}$ be the embedding of items in the same factor space with items representations $\mathbf{x}_i$ as columns. A recommendation is performed on the basis of the ranking induced by the following factorization:

$$\hat{\mathbf{R}} = \mathbf{W}\mathbf{X}$$

The factorization parameters are typically computed minimizing a functional

$$\mathcal{P}(\mathbf{R}, \mathbf{W}, \mathbf{X}) = \mathcal{L}(\mathbf{R}, \mathbf{W}\mathbf{X}) + \mathcal{R}(\mathbf{W}, \mathbf{X})$$

where $\mathcal{L}(\mathbf{R}, \mathbf{W}\mathbf{X})$ is the training loss, that is a measure of the discrepancy between the initial rating matrix and the obtained approximated matrix, and $\mathcal{R}(\mathbf{W}, \mathbf{X})$ is a regularization term.

Now, we review well known state-of-the-art methods in the framework above. We start with user-based NN and item-based NN prediction although in these cases there is a fixed embedding and no real training has to be performed.

User-based NN prediction.

In the user-based NN case the embedding is performed in $\mathbb{R}^n$, the space of users, according to:

$$\mathbf{w}_u^{(v)} = \begin{cases} s_{uv} & \text{if } v \in \text{NN}(u) \\ 0 & \text{otherwise} \end{cases}, \; s_{uv} = \cos(\mathbf{r}_u, \mathbf{r}_v) = \frac{\mathbf{r}_u^\top \mathbf{r}_v}{||\mathbf{r}_u||||\mathbf{r}_v||}$$

$$\mathbf{x}_i^{(v)} = \frac{r_{vi}}{||\mathbf{r}_i||} \quad \text{i.e.} \quad \mathbf{x}_i = \frac{\mathbf{r}_i}{||\mathbf{r}_i||}$$

Item-based NN prediction.

In this case, the embedding is performed in $\mathbb{R}^m$, the space of items, according to:

$$\mathbf{w}_u^{(j)} = \frac{r_{uj}}{||\mathbf{r}_u||} \quad \text{i.e.} \quad \mathbf{w}_u = \frac{\mathbf{r}_u}{||\mathbf{r}_u||}$$

$$\mathbf{x}_i^{(j)} = \begin{cases} s_{ij} & \text{if } j \in \text{NN}(i) \\ 0 & \text{otherwise} \end{cases} \quad \text{and} \quad s_{ij} = \cos(\mathbf{r}_i, \mathbf{r}_j) = \frac{\mathbf{r}_i^\top \mathbf{r}_j}{||\mathbf{r}_i||||\mathbf{r}_j||}$$

Weighted Regularized Matrix Factorization (WR-MF).

In WR-MF, both the user embedding and the item embedding have to be trained. These embeddings are performed in a common space of dimension k, which is an external parameter, and the optimization problem consists of a regularized weighted squared loss

$$\mathcal{P}(\mathbf{R}, \mathbf{W}, \mathbf{X}) = \sum_{u \in \mathbb{U}} \sum_{i \in \mathbb{I}} c_{ui}(\hat{r}_{ui} - 1)^2 + \lambda(||\mathbf{W}||^2 + ||\mathbf{X}||^2)$$

where c_{ui} are apriori given weights for each (rated and unrated) item. The coefficients of positive feedback typically gets a greater value (see [6, 4]).

SLIM (Sparse Linear Method).

The method in [5] can be cast in our framework as well. In this case, the embedding is performed in item space ($k = m$). SLIM learns a sparse matrix $\mathbf{X} \in \mathbb{R}^{m \times m}$ by minimizing a least square loss $LS(\mathbf{R}, \mathbf{X}\mathbf{W})$ such that $x_{ij} \geq 0, x_{ii} = 0$. Specifically, the regularization term is $\mathcal{R}(\mathbf{W}, \mathbf{X}) = \frac{\beta}{2}||\mathbf{X}||_F^2 + \lambda||\mathbf{X}||_1$ where β, λ are two external parameters. The matrix $\mathbf{W}$ is fixed ($\mathbf{W} = \mathbf{R}$).

Bayesian Personalized Ranking (BPR).

The method in [7] learns the two embeddings in a factor space of dimension k (external parameter). The optimization problem is obtained by a Bayesian interpretation of AUC maximization and it is defined by:

$$\mathcal{P}(\mathbf{R}, \mathbf{W}, \mathbf{X}) = \sum_{u \in \mathbb{U}} \sum_{i \in \mathbb{I}} \ln(\sigma(\hat{r}_{ui} - \hat{r}_{uj})) + \lambda_w ||\mathbf{W}||^2 + \lambda_x ||\mathbf{X}||^2$$

where $\sigma(\cdot)$ is the sigmoid function and λ_w, λ_x are external parameters. The problem is clearly not convex and the training is performed by stochastic gradient descent.

4. MARGIN AND AUC OPTIMIZATION

As seen in the previous sections, typical approaches to collaborative filtering cast the problem of item recommendation as a regularized regression problem [4, 6, 5]. This means that they try to fit the matrix $\mathbf{R}$ with a regularized function that should predict values close to 1 for rated items and values close to 0 for unrated items. For the regression case, a natural choice of a loss to optimize is the least square loss. Even if this choice has been shown to be very good for explicit feedback, where the evaluation is usually performed with RMSE, this seems less suited to the implicit feedback setting where different ranking-based evaluation measure, like AUC, mean average precision, etc., are preferred.

In this paper, we propose a different paradigm inspired to preference learning and instance ranking [3] by considering preferences on pairs of items of a user. In particular, we consider a user preferring an item i over an item j ($i \succ_u j$) whenever the user u has rated item i but we have not information about the user preference on item j. Moreover, we assume that users act independently, that is, the fact that a user has ranked or not an item gives no additional information about what other users do. With this hypothesis any user can be modeled independently.

In our model a user ranks items based on the usual scoring function $\hat{r}_{ui} = \mathbf{w}_u^\top \mathbf{x}_i$ but with the additional constraint that the vectors $\mathbf{w}_u$ and $\mathbf{x}_i$ are enforced to be unitary norm vectors. Note that, given this assumption, we have $|\hat{r}_{ui}| \leq 1$ since $|\hat{r}_{ui}| \leq ||\mathbf{w}_u|| \cdot ||\mathbf{x}_i|| = 1$ always holds.

In the following, we focus on a fixed item representation, namely the same used by user-based NN method, i.e. $\mathbf{x}_i = \mathbf{r}_i / ||\mathbf{r}_i||$.

Given a user, we define the *margin* for a item pair (i, j) such that $i \succ_u j$, and we denote it by $\rho(i \succ_u j)$, as in the following:

$$\rho(i \succ_u j) = \frac{1}{2}(\hat{r}_{ui} - \hat{r}_{uj}) = \frac{1}{2}\mathbf{w}_u^\top(\mathbf{x}_i - \mathbf{x}_j).$$

Note that $-1 \leq \rho(i \succ_u j) \leq +1$ holds for any pair of items. The function $\rho(i \succ_u j)$ well approximates the indicator function $[\![\hat{r}_{ui} > \hat{r}_{uj}]\!]$ having value 1 when a pair is correctly ranked or 0 otherwise. In particular, we have that the margin is a non trivial lower bound of the indicator function:

$$-1 \le \rho(i \succ_u j) \le [\![\hat{r}_{ui} > \hat{r}_{uj}]\!] \le +1.$$

Following the same line as in [2] we consider for each user u the following game. Let $\mathcal{P}_{min}$ (the nature), and $\mathcal{P}_{max}$ (the player). On each round of the game, $\mathcal{P}_{max}$ picks an hypothesis $\mathbf{w}_u$ such that $||\mathbf{w}_u|| = 1$, and simultaneously $\mathcal{P}_{min}$ picks a preference $i \succ_u j$. $\mathcal{P}_{max}$ wants to maximize its pay-off defined as the achieved margin on the pair of items which, in our setting, is defined by $\rho(i \succ_u j)$.

Let now be given a mixed strategy for the $\mathcal{P}_{min}$ player defined by two probability distributions: $\boldsymbol{\alpha}_u^+$, the probability of each positive item to be selected, and $\boldsymbol{\alpha}_u^-$ the corresponding probabilities for negative items. We can assume these probabilities can be marginalized as the associated events are independent. In other words, the probability to pick the pair $(i \succ_u j)$ is simply given by $\alpha_{ui}^+\alpha_{uj}^-$. Hence, the value of the game, i.e. the expected margin obtained in a game, will be:

$$\mathbb{E}_{\boldsymbol{\alpha}}[\rho] = \frac{1}{2} \sum_{i \in \mathbb{I}(u), j \notin \mathbb{I}(u)} \alpha_{ui}^+ \alpha_{uj}^- (\mathbf{w}_u^\top \mathbf{x}_i - \mathbf{w}_u^\top \mathbf{x}_j)$$

When the player $\mathcal{P}_{min}$ is free to choose any possible strategy, the following problem is obtained that determines the equilibrium of the game, that is

$$(\mathbf{w}_u^*, \boldsymbol{\alpha}_u^*) = \arg \max_{||\mathbf{w}_u||=1} \min_{\boldsymbol{\alpha} \in \mathbb{C}(u)} \mathbb{E}_{\boldsymbol{\alpha}}[\rho]$$

where $\mathbb{C}(u) = \{\boldsymbol{\alpha} | \alpha_i \ge 0, \sum_{i \in \mathbb{I}(u)} \alpha_i = 1, \sum_{i \notin \mathbb{I}(u)} \alpha_i = 1\}$.

The expected margin can now be reformulated as follows:

$$\begin{aligned}
\mathbb{E}_{\boldsymbol{\alpha}_u}[\rho] &= \frac{1}{2} \sum_{i \in \mathbb{I}(u)} \sum_{j \notin \mathbb{I}(u)} \alpha_{ui}\alpha_{uj}(\hat{r}_{ui} - \hat{r}_{uj}) \\
&= \frac{1}{2}(\sum_{i \in \mathbb{I}(u)} \alpha_{ui}\hat{r}_{ui} \underbrace{\sum_{j \notin \mathbb{I}(u)} \alpha_{uj}}_{=1} - \sum_{j \notin \mathbb{I}(u)} \alpha_{uj}\hat{r}_{uj} \underbrace{\sum_{i \in \mathbb{I}(u)} \alpha_{ui}}_{=1}) \\
&= \frac{1}{2} \sum_{i \in \mathbb{I}} y_{ui}\alpha_{ui}\hat{r}_{ui} = \frac{1}{2} \sum_{i \in \mathbb{I}} y_{ui}\alpha_{ui}\mathbf{w}_u^\top \mathbf{x}_i \\
&= \frac{1}{2}\mathbf{w}_u^\top \mathbf{X}\mathbf{Y}_u\boldsymbol{\alpha}_u
\end{aligned}$$

where we set

$$\mathbf{y}_u : y_{ui} = \begin{cases} +1 & \text{if } i \in \mathbb{I}(u) \\ -1 & \text{if } i \notin \mathbb{I}(u) \end{cases} \quad \text{and } \mathbf{Y}_u = \text{diag}(\mathbf{y}_u).$$

Setting $\mathbf{v}_u = \mathbf{X}\mathbf{Y}_u\boldsymbol{\alpha}_u$, then the $\mathbf{w}_u$ maximizing the expected margin is simply defined by $\mathbf{w}_u^* = c\mathbf{v}_u$, where $c = ||\mathbf{v}_u||^{-1}$. Note that, since c depends on the user only, then the ranking of items induced by $\mathbf{w}_u^*$ is the same as the one induced by $\mathbf{v}_u$. Substituting back the obtained solution for $\mathbf{w}_u$, we obtain:

$$\begin{aligned}
\mathbb{E}_{\boldsymbol{\alpha}_u}[\rho] &= \frac{1}{2} \sum_{i \in \mathbb{I}} y_{ui}\alpha_{ui}\mathbf{w}_u^\top \mathbf{x}_i \\
&= \frac{1}{2||\mathbf{v}_u||} \sum_{i \in \mathbb{I}} y_{ui}\alpha_{ui}\mathbf{v}_u^\top \mathbf{x}_i \\
&= \frac{\mathbf{v}_u^\top \mathbf{v}_u}{2||\mathbf{v}_u||} = \frac{1}{2}||\mathbf{v}_u||
\end{aligned}$$

Every element of the set $\mathbb{C}(u)$ represents two points, one in the convex hull of positive items, that is $\mathbf{x}^+ = \sum_{i \in \mathbb{I}(u)} \alpha_{ui}\mathbf{x}_i$, and another in the convex hull of negative items, that is $\mathbf{x}^- = \sum_{j \notin \mathbb{I}(u)} \alpha_{uj}\mathbf{x}_j$, for any given user. It is possible to show that $\mathbf{v}_u$ corresponds to the vector starting from $\mathbf{x}^-$ and ending in $\mathbf{x}^+$, that is $\mathbf{v}_u = \mathbf{x}^+ - \mathbf{x}^-$. As shown above, the expected margin is proportional to the length of this vector.

Given the game defined above, we see now that the expectation of the margin is also a lower bound of the expected AUC obtained in the game for that user. In fact, given the probability distributions $\boldsymbol{\alpha} \in \mathbb{C}(u)$ we get:

$$\begin{aligned}
\text{AUC}_{\alpha}(u) &= \sum_{i \in \mathbb{I}(u), j \notin \mathbb{I}(u)} \alpha_{ui}\alpha_{uj}[\![\hat{r}_{ui} > \hat{r}_{uj}]\!] \\
&\ge \sum_{i \in \mathbb{I}(u), j \notin \mathbb{I}(u)} \alpha_{ui}\alpha_{uj}\rho(i \succ_u j) = \mathbb{E}_{\boldsymbol{\alpha}_u}[\rho]
\end{aligned}$$

As discussed in [2] for the classification case, the pure maximization of the minimum margin is not necessarily the best choice. Especially in our case, noise is always present in the negative set of items. For this, we consider two quadratic regularization terms that give a bias towards uniform solutions for $\boldsymbol{\alpha}$, namely $\alpha_{ui} = 1/m_u^+$ whenever $i \in \mathbb{I}(u)$ and $\alpha_{uj} = 1/m_u^-$ whenever $j \notin \mathbb{I}(u)$. In the experiments we will see that an imbalanced setting of these regularization parameters is crucial for the effectiveness of our method.

For each user, we define the following optimization problem with the aim to maximize the expected margin and the AUC lower bound:

$$\min_{\boldsymbol{\alpha}_u \in \mathbb{C}(u)} \mathcal{Q}(\boldsymbol{\alpha}_u) + \mathcal{R}(\boldsymbol{\alpha}_u)$$

where

$$\mathcal{Q}(\boldsymbol{\alpha}_u) = ||\mathbf{v}_u||^2 = \boldsymbol{\alpha}_u^\top \mathbf{Y}_u \mathbf{X}^\top \mathbf{X} \mathbf{Y}_u \boldsymbol{\alpha}_u$$

$$\mathcal{R}(\boldsymbol{\alpha}_u) = \lambda_p \sum_{i \in \mathbb{I}(u)} \alpha_{ui}^2 + \lambda_n \sum_{i \notin \mathbb{I}(u)} \alpha_{ui}^2$$

Rearranging the terms in matrix form we obtain:

$$\boldsymbol{\alpha}_u^* = \arg \min_{\boldsymbol{\alpha}_u \in \mathbb{C}(u)} \boldsymbol{\alpha}_u^\top \left(\mathbf{Y}_u \mathbf{X}^\top \mathbf{X} \mathbf{Y}_u + \boldsymbol{\Lambda} \right) \boldsymbol{\alpha}_u$$

where the diagonal matrix $\boldsymbol{\Lambda}$ is defined as follows:

$$\boldsymbol{\Lambda} : \boldsymbol{\Lambda}(i,j) = \begin{cases} \lambda_p & \text{if } i = j \text{ and } i \in \mathbb{I}(u) \\ \lambda_n & \text{if } i = j \text{ and } i \notin \mathbb{I}(u) \\ 0 & \text{otherwise} \end{cases}$$

This problem is convex quadratic and can be optimized with any tool for convex optimization. For our experiments, we used Python and the CVXOPT package[1].

Note that, when $\lambda_p, \lambda_n = +\infty$, it is possible to show that the optimization problem has a closed form solution, that is

$$\mathbf{w}_u \propto \frac{1}{m_u^+} \sum_{i \in \mathbb{I}(u)} \mathbf{x}_i - \frac{1}{m_u^-} \sum_{j \notin \mathbb{I}(u)} \mathbf{x}_j.$$

5. EXPERIMENTS

The datasets used for the evaluation have been extracted from MovieLens1M (ML1M), NetFlixSmall (NFS), BookCrossing (BX). All these datasets correspond to multi-value ratings that we converted into binary ratings (rated/non-rated). In particular we used the following three datasets[2]:

[1] M. S. Andersen, J. Dahl, and L. Vandenberghe. CVXOPT: A Python package for convex optimization, version 1.1.6. Available at cvxopt.org, 2013.

[2] http://www.math.unipd.it/~aiolli/DATA/RECSYS14/

Dataset	Our Method	WRMF
ML1M (3850,2273)	**0.89886**	0.88630
NFS (2979,2544)	**0.89550**	0.87309
BX (1609,1629)	**0.74200**	0.73932

Table 1: AUC comparisons. Our method vs. WRMF (in parenthesis the number of users and items of the subset).

$\lambda_p \setminus \lambda_n$	1	10	100	1000	10000
0.01	0.87751	0.89714	0.89768	0.89631	0.89612
0.1	0.87809	0.89762	0.89792	0.89646	0.89626
1	0.87961	**0.89886**	0.89779	0.89584	0.89557
10	0.87926	0.89764	0.89376	0.89065	0.89023
100	0.87897	0.89692	0.89178	0.88813	0.88763
1000	0.87903	0.89685	0.89151	0.88777	0.88727

Table 2: ML1M AUC results varying the external parameters

1. **ML1M**. The ML1M[3] dataset is a subset of the MovieLens1M movie-related data corresponding to users who rated from 20 to 500 items and items being rated by from 20 to 500 users.

2. **NFS**. The NFS dataset[4] is a subset of the NetFlixSmall movie-related data corresponding to users who rated from 20 to 500 items and items being rated by from 5 to 500 users.

3. **BX**. The BX[5] dataset is a subset of the BookCrossing book-related data corresponding to users who rated from 10 to 500 items and items being rated by from 10 to 500 users.

To evaluate the performance of the proposed method, and compare it with other methods, 5-fold cross-validation has been employed. We first compared all the state-of-the-art methods described in this paper (UserBased and ItemBased NN, WRMF, SLIM, and BPR). MyMediaLite[6] implementations of these methods has been used. WRMF performed similar to BPR and better than SLIM and it has been selected for direct comparison with our method (see Table 1). Note that this partially contradicts results presented in the original paper [5] where they were presented in terms of HR (leave-one-out). Worse performance of SLIM can be due to the fact that it is based on least-square/regression and does not weights differently positive (unambiguous) and negative (ambiguous) implicit feedback. From the table we can note that our method outperforms the state-of-the-art method in terms of AUC. We observed the same trend when using other top-N measures (e.g. map@10).

$\lambda_p \setminus \lambda_n$	1	10	100	1000	10000
0.01	0.85428	0.88473	0.89414	0.89466	0.89469
0.1	0.85400	0.88490	0.89440	0.89489	0.89492
1	0.85396	0.88590	0.89507	0.89547	**0.89550**
10	0.85220	0.88412	0.89388	0.89416	0.89415
100	0.85159	0.88357	0.89319	0.89469	0.89337
1000	0.85290	0.88358	0.89309	0.89327	0.89326

Table 3: NFS AUC results varying the external parameters

$\lambda_p \setminus \lambda_n$	1	10	100	1000	10000
0.01	0.69457	0.72075	0.73837	0.74087	0.74114
0.1	0.69483	0.72106	0.73867	0.74112	0.74175
1	0.69473	0.72165	0.73925	0.74175	**0.74200**
10	0.69244	0.71965	0.73751	0.74012	0.74039
100	0.69179	0.71893	0.73683	0.73939	0.73966
1000	0.69184	0.71897	0.73676	0.73931	0.73957

Table 4: BX AUC results varying the external parameters

Further analysis has been done to demonstrate the robustness of the proposed method with respect to the setting of external parameters. In Tables 2,3,4, the results obtained with different settings show that $\lambda_p \approx 1$, $\lambda_n \approx 1000$ always give good results and this is a great advantage of our method with respect to other state-of-the-art methods that typically are more sensitive to the external parameter setting and often require heavy parameter tuning to work well.

Finally, we can note that the results of our method tend to get worse for big λ_p values or small λ_n values. In the case of λ_p (unambiguous information parameter), smaller values allows the model to (safely) optimize the minimum margin in the training set. In the case of λ_n, small values tend to make the ambiguous information to be too relevant in the construction of the model and this introduces erroneous information in the final model estimate. On the other side, larger λ_n values tend to mitigate the negative effect of ambiguous information on the constructed model and this explains why so high values of λ_n are preferable in our context.

6. CONCLUSIONS AND FUTURE WORK

We presented an effective method for AUC optimization of top-N recommendation with implicit feedback. Future work will extend the method to learn in a principled way the user and item embeddings simultaneously.

7. REFERENCES

[1] F. Aiolli. Efficient top-n recommendation for very large scale binary rated datasets. In Q. Yang, I. King, Q. Li, P. Pu, and G. Karypis, editors, *RecSys*, pages 273–280. ACM, 2013.

[2] F. Aiolli, G. D. S. Martino, and A. Sperduti. A kernel method for the optimization of the margin distribution. In *ICANN (1)*, pages 305–314, 2008.

[3] F. Aiolli and A. Sperduti. A preference optimization based unifying framework for supervised learning problems. In J. Fürnkranz and E. Hüllermeier, editors, *Preference Learning*, pages 19–42. Springer-Verlag, 2010.

[4] Y. Hu, Y. Koren, and C. Volinsky. Collaborative filtering for implicit feedback datasets. In *ICDM*, pages 263–272, 2008.

[5] X. Ning and G. Karypis. Slim: Sparse linear methods for top-n recommender systems. In *ICDM*, pages 497–506, 2011.

[6] R. Pan, Y. Zhou, B. Cao, N. N. Liu, R. Lukose, M. Scholz, and Q. Yang. One-class collaborative filtering. In *ICDM*, 2008.

[7] S. Rendle, C. Freudenthaler, Z. Gantner, and L. Schmidt-Thieme. Bpr: Bayesian personalized ranking from implicit feedback. In *Proc. of the 25th Conference on Uncertainty in Artificial Intelligence (UAI)*, 2009.

[3] http://grouplens.org/datasets/movielens/

[4] http://www.netflixprize.com/

[5] http://www.informatik.uni-freiburg.de/~cziegler/BX/

[6] http://mymedialite.net/