Collaborative Filtering via Different Preference Structures

Shaowu Liu[1], Na Pang[2] Guandong Xu[1], and Huan Liu[3]

[1] University of Technology Sydney, Australia
[2] School of Cyber Security, University of Chinese Academy of Sciences, China
[3] Arizona State University, USA
shaowu.liu@uts.edu.au, pangna@iie.ac.cn, guandong.xu@uts.edu.au, huan.liu@asu.edu

Abstract. Recently, social network websites start to provide third-parity sign-in options via the OAuth 2.0 protocol. For example, users can login *Netflix* website using their *Facebook* accounts. By using this service, accounts of the same user are linked together, and so does their information. This fact provides an opportunity of creating more complete profiles of users, leading to improved recommender systems. However, user opinions distributed over different platforms are in different preference structures, such as ratings, rankings, pairwise comparisons, voting, etc. As existing collaborative filtering techniques assume the homogeneity of preference structure, it remains a challenge task of how to learn from different preference structures simultaneously. In this paper, we propose a fuzzy preference relation-based approach to enable collaborative filtering via different preference structures. Experiment results on public datasets demonstrate that our approach can effectively learn from different preference structures, and show strong resistance to noises and biases introduced by cross-structure preference learning.

Keywords: Recommender System, Pairwise Preference, Data Mining

1 Introduction

Personalized recommendation is an important component of today's business. By observing user behaviors, recommender systems can identify potential users of a product, or products that could be interested by a targeted user. An important technique to make recommendations is collaborative filtering (CF). CF is based on the intuition that there exist shared patterns to transfer preferences across like-minded users. For example, whether a targeted user will like a movie can be inferred by other users who have similar taste to the targeted user. The taste of a user can be extracted from user preferences in different structures, such as ratings [7], rankings [8], pairwise comparisons [5], voting [11], text reviews, etc. A common assumption made by CF is the homogeneity of preference structures, where only one type of preference structure is accepted at a time.

The last decade has seen a growing trend towards creating and managing more profiles in social network, such as *Facebook*, *LinkedIn*, *Netflix*, etc. Furthermore, the popularization of third-party sign-in via the OAuth 2.0 protocol has made it possible to link multiple profiles of the same user together. In light of this trend, it becomes possible to alleviate the *cold-start* problem by learning user preferences from multiple profiles, e.g., a new user of *Netflix* may have been used *Facebook* for a while. Nevertheless, user preferences collected from different platforms are often expressed in different preference structures. For example, 5-star rating is used by *Netflix*, but voting (thumbs up) is used by *Facebook*. Despite of explicit preferences, additional complexity is added if implicit preferences such as *page views* and *mouse clicks* are also taken into consideration.

Moreover, user preferences collected from different platforms may contain different noises and biases, as the user preferences not only reflect inherent quality of the product but also quality of the product providers. For example, a user may rate a movie 3 star on one platform, but 5 star for the same movie on another platform due to 3D support, which is called misattribution of memory [13] in psychology. Nevertheless, different preference structures need to be placed on the same scale for accurate discovering of shared patterns to achieve quality recommendations.

In this work, we propose a fuzzy preference relations-based approach to learn from different preference structures. Rather than trying to learn an independent model for each type of preference structure, we propose to *simultaneously learn user preferences in all structures in one model*. With the assistance of *PR*, user preferences in different forms can be fused seamlessly. For example, user preferences expressed as 5-star ratings, binary ratings, and *page views* can not be directly fused in general. However, all those user preferences can be deduced into the *PR* format by performing pairwise comparison on items. Once the user preferences are represented in *PR*, a direct merge can be performed. In fact, converting user preferences into *PR* not only provides a method to merge heterogeneous data but also reduces the biases that come with heterogeneity, i.e., the relative ordering of items is resistant to biases. *The main contribution of this work is proposing an approach to learn from multiple data sources with different preference structures such as ratings, page views, mouse clicks, reviews, etc.*

The rest of the paper is organized as follows. Section 2 introduces the basic concepts of CF and preference structures. Section 3 is devoted to describe the proposed method. In Section 4, the proposed method is applied to public datasets for top-N recommendation. Finally, conclusions are drawn in Section 5.

2 Preliminaries and Related Work

This section briefly summarizes necessary background related to the *heterogeneous sources* problem and the *preference relations* that form the basis of our solution.

2.1 Heterogeneous Sources

User preferences are usually assumed to come from a single *homogeneous source*. This assumption is becoming invalid given the rapid development of online social networks in which users maintain multiple profiles and the form of preferences diverges. We define two sources as heterogeneous if their preferences are 1) in different forms, e.g., *ratings* and *clicks*; 2) in different scales, e.g., *5-star* scale and *6-star* scale; 3) or biased differently due to factors irrelevant to the items' quality, e.g., quality of the service providers. Based on this definition, not only the physically separated sources are heterogeneous but a source changed significantly is also considered heterogeneous to itself.

In general, user preferences from heterogeneous sources cannot be merged directly as they may be in different forms. Even if their forms are the same, the scales could be different, where a force casting may change the meaning of preferences. In case that the scales are the same, biases are still introduced by the sources which make the recommendations inaccurate.

2.2 Preference Relation

Preference relation (PR) encodes user preferences in the form of *relative* ordering between items, which is a useful alternative representation to *absolute* ratings as suggested in recent works [3,5,9]. In fact, existing preferences such as ratings or other types of preferences can be easily represented as *PR* and then merged into a single dataset as shown in Fig. 1. . This property is particularly useful for the *cold-start* problem but has been overlooked in literature.

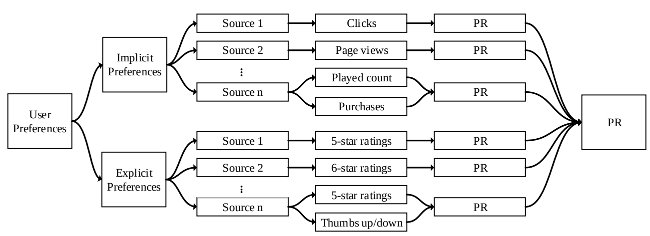


Fig. 1: Flow from user preferences to PR

We formally define the *PR* as follows. Let $\mathcal{U} = \{u\}^n$ and $\mathcal{I} = \{i\}^m$ denote the set of n users and m items, respectively. The *PR* of a user $u \in \mathcal{U}$ between items i and j is encoded as π_{uij}, which indicates the strength of user u's preference relation for the ordered item pair (i, j). A higher value of π_{uij} indicates a stronger preference to the first item over the second item.

The preference relation is defined as

$$\pi_{uij} = \begin{cases} (\frac{2}{3}, 1] & \text{if } i \succ j \ (u \text{ prefers } i \text{ over } j) \\ [\frac{1}{3}, \frac{2}{3}] & \text{if } i \simeq j \ (i \text{ and } j \text{ are equally preferable}) \\ [0, \frac{1}{3}) & \text{if } i \prec j \ (u \text{ prefers } j \text{ over } i) \end{cases} \tag{1}$$

where $\pi_{uij} \in [0, 1]$ and $\pi_{uij} = 1 - \pi_{uji}$.

An interval is allocated for each preference category, i.e., *preferred*, *equally preferred*, and *less preferred*. Indeed, each preference category can be further break down into more intervals, though here in this paper we consider the minimal case of 3 intervals.

Similar to [3], the *PR* can be converted into *user-wise preferences* over items which encode the ranking of items evaluated by a particular user. The user-wise preference is defined as

$$p_{ui} = \frac{\sum_{j \in \mathcal{I}_u \setminus i} [\![\pi_{uij} > \frac{2}{3}]\!] - \sum_{j \in \mathcal{I}_u \setminus i} [\![\pi_{uij} < \frac{1}{3}]\!]}{|\Pi_{ui}|} \tag{2}$$

where $[\![\cdot]\!]$ gives 1 for *true* and 0 for *false*, and Π_{ui} is the set of user u's PR related to item i. The user-wise preference p_{ui} falls in the interval $[-1, 1]$, where 1 and -1 indicate that item i is the most and the least preferred item for u, respectively.

3 Preference Relation-based Conditional Random Fields

In this section, we propose the *Preference Relation-based Conditional Random Fields* (PrefCRF) to model both the heterogeneous preferences and the side information. The rest of this section defines the *PR*-based RecSys problem, and introduces the concept of the *PrefNMF* [5] that forms our underlying model, followed by a detailed description of the *PrefCRF* and discussion on issues such as feature design, parameter estimation, and predictions.

3.1 Problem Statement

Generally, the task of *PR*-based RecSys is to take *PR* as input and output Top-N recommendations. Specifically, let $\pi_{uij} \in \Pi$ encode the *PR* of each user $u \in \mathcal{U}$, and each π_{uij} is defined over an ordered item pair (i, j), denoting $i \prec j$, $i \simeq j$, or $i \succ j$. The main task towards Top-N recommendations is to estimate the value of each unknown $\pi_{uij} \in \Pi_{unknown}$, such that $\hat{\pi}_{uij}$ approximates π_{uij}. This can be considered as an optimization task that performs directly on the *PR*

$$\hat{\pi}_{uij} = \underset{\hat{\pi}_{uij} \in [0,1]}{\arg\min} \left(\pi_{uij} - \hat{\pi}_{uij}\right)^2 \tag{3}$$

However, it can be easier to estimate the $\hat{\pi}_{uij}$ by the difference between two user-wise preferences p_{ui} and p_{uj}, i.e., $\hat{\pi}_{uij} = \phi(\hat{p}_{ui} - \hat{p}_{uj})$, where $\phi(\cdot)$ is a function that bounds the value into $[0, 1]$ and ensures $\phi(0) = 0.5$. For example, the *inverse-logit* function $\phi(x) = \frac{e^x}{1+e^x}$ can be used when user-wise preferences involve large values. The objective of this paper is then to solve the following optimization problem

$$(\hat{p}_{ui}, \hat{p}_{uj}) = \underset{\hat{p}_{ui}, \hat{p}_{uj}}{\arg\min} (\pi_{uij} - \phi(\hat{p}_{ui} - \hat{p}_{uj}))^2 \tag{4}$$

which optimizes the user-wise preferences directly, and Top-N recommendations can be obtained by simply sorting the estimated user-wise preferences.

3.2 Preference Relation-based Matrix Factorization

Matrix Factorization (MF) [7] is a popular *RecSys* approach that has mainly been applied to absolute ratings. Recently, the *PrefNMF* [5] model was proposed to accommodate *PR* input for *MF* models. Like traditional *MF* models, the *PrefNMF* model discovers the latent factor space shared between users and items, where the latent factors describe both the *taste* of users and the *characteristics* of items. The attractiveness of an item to a user is then measured by the inner product of their latent feature vectors.

Formally, each user u is associated with a latent feature vector $\mathbf{u}_u \in \mathbb{R}^k$, and each item i is associated with a latent feature vector $\mathbf{v}_i \in \mathbb{R}^k$, where k is the dimension of the latent factor space. The attractiveness of items i and j to user u are $\mathbf{u}_u^\top \mathbf{v}_i$ and $\mathbf{u}_u^\top \mathbf{v}_j$, respectively. When $\mathbf{u}_u^\top \mathbf{v}_i > \mathbf{u}_u^\top \mathbf{v}_j$, the item i is said to be more preferable to the user u than item j, i.e., $i \succ j$. The strength of this preference relation π_{uij} can be estimated by $\mathbf{u}_u^\top(\mathbf{v}_i - \mathbf{v}_j)$, and the *inverse-logit* function is applied to ensure $\hat{\pi}_{uij} \in [0, 1]$: $\hat{\pi}_{uij} = \frac{e^{\mathbf{u}_u^\top(\mathbf{v}_i - \mathbf{v}_j)}}{1+e^{\mathbf{u}_u^\top(\mathbf{v}_i - \mathbf{v}_j)}}$.

The latent feature vectors $\mathbf{u}_u$ and $\mathbf{v}_i$ are learned by minimizing regularized squared error with respect to the set of all known preference relations Π:

$$\min_{\mathbf{u}_u, \mathbf{v}_i \in \mathbb{R}^k} \sum_{\pi_{uij} \in \Pi \wedge (i<j)} (\pi_{uij} - \hat{\pi}_{uij})^2 + \lambda(\|\mathbf{u}_u\|^2 + \|\mathbf{v}_i\|^2) \tag{5}$$

where λ is the regularization coefficient.

3.3 Conditional Random Fields

Conditional Random Fields (CRF) [14] model a set of random variables having Markov property with respect to an undirected graph $\mathcal{G}$, and each random variable can be conditioned on a set of global observations $\mathbf{o}$. The undirected graph $\mathcal{G}$ consists of a set of vertexes $\mathcal{V}$ connected by a set of edges $\mathcal{E}$ without orientation, where two vertexes are neighboring to each other when connected. Each vertex in $\mathcal{V}$ encodes a random variable, and the Markov property implies that a variable is conditionally independent of others given its neighbors.

In this work, we use *CRF* to model interactions among user-wise preferences conditioned on side information with respect to a set of undirected graphs. Specifically for each user u, there is a graph $\mathcal{G}_u$ with a set of vertexes $\mathcal{V}_u$ and a set of edges $\mathcal{E}_u$. Each vertex in $\mathcal{V}_u$ represents a user-wise preference p_{ui} of user u on the item i. Each edge in $\mathcal{E}_u$ captures a relation between two preferences by the same user.

Each vertex is conditioned on a set of global observations $\mathbf{o}$, which is the *side information* in our context. Specifically, each user u is associated with a set of L attributes $\{\mathbf{o}_u\}^L$ such as *age*, *gender* and *occupation*. Similarly, each item i is associated with a set of M attributes $\{\mathbf{o}_i\}^M$ such as *genres* for movie. Those side information is encoded as the set of global observations $\mathbf{o} = \{\{\mathbf{o}_u\}^L, \{\mathbf{o}_i\}^M\}$.

Formally, let $\mathbf{p}_u = \{p_{ui} \mid i \in \mathcal{I}_u\}$ be the joint set of preferences expressed by user u, then we are interested in modeling the conditional distribution $P(\mathbf{p}_u \mid \mathbf{o})$ over the graph $\mathcal{G}_u$.

$$P(\mathbf{p}_u \mid \mathbf{o}) = \frac{1}{Z_u} \Psi_u(\mathbf{p}_u, \mathbf{o}) \tag{6}$$

$$\Psi_u(\mathbf{p}_u, \mathbf{o}) = \prod_{(ui) \in \mathcal{V}_u} \psi_{ui}(p_{ui}, \mathbf{o}) \prod_{(ui,uj) \in \mathcal{E}_u} \psi_{ij}(p_{ui}, p_{uj}) \tag{7}$$

where $Z_u(\mathbf{o})$ does normalization to ensure $\sum_{\mathbf{p}_u} P(\mathbf{p}_u \mid \mathbf{o}) = 1$, and $\psi(\cdot)$ is a positive function known as *potential*. The potential $\psi_{ui}(\cdot)$ captures the global observations associated to the user u and the item i, and the potential $\psi_{ij}(\cdot)$ captures the correlations between two preferences p_{ui} and p_{uj}

$$\psi_{ui}(p_{ui}, \mathbf{o}) = \exp\{\mathbf{w}_u^\top \mathbf{f}_u(p_{ui}, \mathbf{o}_i) + \mathbf{w}_i^\top \mathbf{f}_i(p_{ui}, \mathbf{o}_u))\} \tag{8}$$

$$\psi_{ij}(p_{ui}, p_{uj}) = \exp\{w_{ij} f_{ij}(p_{ui}, p_{uj})\} \tag{9}$$

where $\mathbf{f}_u$, $\mathbf{f}_i$, and f_{ij} are the features to be designed shortly in Section 3.4, and $\mathbf{w}_u$, $\mathbf{w}_i$, and w_{ij} are the corresponding weights realizing the importance of each feature. With the weights estimated from data, the unknown preference p_{ui} can be predicted as

$$\hat{p}_{ui} = \operatorname*{arg\,max}_{p_{ui} \in [-1,1]} P(p_{ui} \mid \mathbf{p}_u, \mathbf{o}) \tag{10}$$

where $P(p_{ui} \mid \mathbf{p}_u, \mathbf{o})$ measures the prediction confidence.

The *Ordinal Logistic Regression* [10] is then used to convert the user-wise preferences p_{ui} into ordinal values, which assumes that the preference p_{ui} is chosen based on the interval to which the latent utility belongs:

$$p_{ui} = l \text{ if } x_{ui} \in (\theta_{l-1}, \theta_l] \text{ and } p_{ui} = L \text{ if } x_{ui} > \theta_{L-1} \tag{11}$$

where L is the number of ordinal levels and θ_l are the threshold values of interest. The probability of receiving a preference l is therefore:

$$Q(p_{ui} = l \mid u, i) = \int_{\theta_{l-1}}^{\theta_l} P(x_{ui} \mid \theta) \, \mathrm{d}\theta = F(\theta_l) - F(\theta_{l-1}) \tag{12}$$

where $F(\theta_l)$ is the cumulative logistic distribution evaluated at θ_l.

3.4 PrefCRF: Unifying PrefNMF and CRF

The *CRF* provides a principled way of capturing both the side information and interactions among preferences. However, it employs the log-linear modeling as shown in Eq. 7, and therefore does not enable a simple treatment of *PR*. The *PrefNMF*, on the other hand, accepts *PR* but is weak in utilizing side information. The complementary between these two techniques calls for an unified *PrefCRF* model to take all the advantages.

Unification Essentially, the proposed *PrefCRF* model captures the side information and promotes the agreement between the *PrefNMF* and the *CRF*. Specifically, the *PrefCRF* model combines the item-item correlations (Eq. 9) and the ordinal distributions $Q(p_{ui} \mid u, i)$ over user-wise preferences obtained from Eq. 12.

$$P(\mathbf{p}_u \mid \mathbf{o}) \propto \Psi_u(\mathbf{p}_u, \mathbf{o}) \prod_{p_{ui} \in \mathbf{p}_u} Q(p_{ui} \mid u, i) \tag{13}$$

where Ψ_u is the potential function capturing the side information and interaction among preferences related to user u. Though there is a separated graph for each user, the weights are optimized across all graphs.

Feature Design A feature is essentially a function f of $n > 1$ arguments that maps the n-dimensional input into the unit interval $f : \mathbb{R}^n \to [0, 1]$. We design the following kinds of features:

Correlation Features The item-item correlation is captured by the feature

$$f_{ij}(p_{ui}, p_{uj}) = g(|(p_{ui} - \bar{p}_i) - (p_{uj} - \bar{p}_j)|) \tag{14}$$

where $g(\alpha)$ normalizes feature values and α plays the role of deviation, and $\bar{p}_i$ and $\bar{p}_j$ are the average user-wise preference for items i and j, respectively.

Attribute Features Each user u and item i has a set of attributes $\mathbf{o}_u$ and $\mathbf{o}_i$, respectively. These attributes are mapped to preferences by the following features

$$\begin{aligned} \mathbf{f}_i(p_{ui}) &= \mathbf{o}_u g(|(p_{ui} - \bar{p}_i)|) \\ \mathbf{f}_u(p_{ui}) &= \mathbf{o}_i g(|(p_{ui} - \bar{p}_u)|) \end{aligned} \tag{15}$$

where $\mathbf{f}_i$ models which users like the item i and $\mathbf{f}_u$ models which classes of items the user u likes.

Since one correlation feature exists for each pair of co-rated items, the number of correlation features can be large, and makes the estimation slow to converge and less robust. Therefore, we only keep strong correlation features $\mathbf{f}_{\text{strong}}$ extracted based on the *Pearson* correlation between items using a user-specified *minimum correlation threshold*.

Parameter Estimation In general, *CRF* models cannot be determined by standard maximum likelihood estimations, instead, approximation techniques are used in practice. This study employs the pseudo-likelihood [1] to estimate parameters by maximizing the regularized sum of log local likelihoods:

$$log\mathcal{L}(\mathbf{w}) = \sum_{p_{ui} \in \Pi} \log P(p_{ui} \mid \mathbf{p}_u, \mathbf{o}) - \frac{1}{2\sigma^2} \mathbf{w}^\top \mathbf{w} \tag{16}$$

where $\mathbf{w}$ are the weights and $1/2\sigma^2$ controls the regularization. To optimize the parameters, we use the stochastic gradient ascent procedure.

Item Recommendation The *PrefCRF* produces distributions over the user-wise preferences, which can be converted into point estimates by computing the expectation

$$\hat{p}_{ui} = \sum_{p_{ui}=l_{min}}^{l_{max}} p_{ui} P(p_{ui} \mid \mathbf{p}_u, \mathbf{o}) \tag{17}$$

where l refers to the intervals of user-wise preferences: from the least to the most preferred. Given the predicted user-wise preferences, the items can be sorted and ranked accordingly.

4 Experiment and Analysis

To study the performance of the proposed *PrefCRF* model, comparisons were done with the following representative algorithms: *KNN* [12], *NMF* [7], *PrefKNN* [3], and *PrefNMF* [5]. We employ two evaluation metrics *Normalized Cumulative Discounted Gain*@T (NDCG@T) [6] that is popular in academia, and *Mean Average Precision*@T (MAP@T) [4] that is common in contests.

4.1 Experimental Settings

Datasets and Experiment Design Experiments are conducted on four public datasets: *MovieLens*-1M [4], *Amazon Movie Reviews* [5], *EachMovie* [6], and *MovieLens*-20M [4]. These datasets or their subsets are transformed to simulate four scenarios of heterogeneous data:

Side Information The impact of side information is studied on the *MovieLens*-1M dataset which provides *gender*, *age*, and *occupation* information about users and *genres* of movies. The dataset contains more than 1 million ratings by $6,040$ users on $3,900$ movies. For a reliable comparison, the dataset is split into training and test sets with different sparsities.

Different Forms *Amazon Movie Reviews* dataset contains two forms of preferences: *textual reviews* and 5-star *ratings*. We extracted a dense subset by randomly selecting 5141 items with at least 60 reviews for each, and 2000 users with at least 60 movies reviews for each, and this results in 271K ratings. For each user, 50 random reviews are selected for training, and the rest are put aside for testing. The training set is further split into half ratings and half textual reviews. Rating-based models are trained on the ratings only, where *PR*-based models utilize textual reviews as well.

Different Scales *EachMovie* dataset contains ratings in 6-star scale that can be easily converted into binary scale, i.e., ratings $1-3$ and $4-6$ are mapped to 0 and 1 respectively. We extract a subset by randomly selecting 3000 users who have rated at least 70 items as a dense dataset is required for splitting.

[4] `http://grouplens.org/datasets/movielens`
[5] `http://snap.stanford.edu/data/web-Movies.html`
[6] `http://grouplens.org/datasets/eachmovie`

The resultant dataset contains 120K ratings on 1495 items. For each user we randomly select 60 ratings for training and leave the rest for testing, and half of the ratings in the training set are mapped into binary scale. Rating-based models are trained on the 6-star ratings while *PR*-based models will exploit the binary ratings as well.

Different Biases We study the impact of biases by adding biases into a stable dataset with minimal existing biases. To prepare such dataset we extract a stable subset from the latest *MovieLens*-20M released on April-2015. Specifically, 258K ratings by 2020 users on 4408 movies released between 2010 and 2015 are extracted, where each user has rated at least 60 ratings. Biases are then introduced by adding a different *Laplace noise* sampled from $Laplace(0, b)$ to each user and item.

For *PR*-based methods, the same conversion method as in [5] is used to converted ratings into *PR*. For example, 1, 0 and 0.5 are assigned to the preference relation π_{uij} when $p_{ui} > p_{uj}$, $p_{ui} < p_{uj}$, and $p_{ui} = p_{uj}$, respectively.

Parameter Setting For a fair comparison, we fix the number of latent factors to 50 for all algorithms. The number of neighbors for *KNN* algorithms is set to 50. We vary the minimum correlation threshold for the *PrefCRF* to examine the performance with different number of features. Different values of regularization coefficient are also tested.

4.2 Results and Analysis

Algorithms are compared on four heterogeneous scenarios: *side information*, *different forms*, *different scales* and *different biases*. The impact of sparsity levels and parameters is studied on the *MovieLens*-1M dataset, while these settings for other experiments are fixed. Each experiment is repeated ten times with different random seeds and we report the mean results with standard deviations. For each experiment, we also performed a paired t-test (two-tailed) with a significance level of 95% on the best and the second best results, and all p-values are less than 1×10^{-5}.

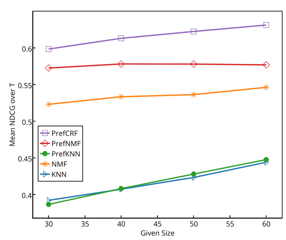
Fusing Side Information Table 1 shows the *NDCG* and *MAP* metrics on Top-N recommendation tasks by compared algorithms. It can be observed that the proposed *PrefCRF*, which captures the side information, consistently outperforms others. To confirm the improvement, we plot the results in Fig. 2b by varying the position T. The figure shows that *PrefCRF* not only outperforms others but has a strong emphasize on top items, i.e., $T < 5$.

The impact of sparsity is investigated by plotting the results against sparsity levels as in Fig. 2a. We can observe that the performance of *PrefCRF* increases linearly given more training data, while its underlying *PrefNMF* model is less extensible to denser dataset.

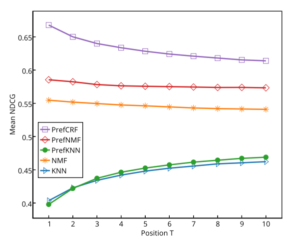
Table 1: Mean results and standard deviation over ten runs on *MovieLens*-1M dataset.

	Given 30				Given 40			
Algorithm	NDCG@1	NDCG@10	MAP@1	MAP@10	NDCG@1	NDCG@10	MAP@1	MAP@10
UserKNN	0.4306 ± 0.0011	0.4081 ± 0.0029	0.3539 ± 0.0071	0.2744 ± 0.0025	0.3695 ± 0.0048	0.4252 ± 0.0036	0.3663 ± 0.0047	0.2877 ± 0.0034
NMF	0.5274 ± 0.0084	0.5195 ± 0.0040	0.5225 ± 0.0081	0.3549 ± 0.0037	0.5424 ± 0.0067	0.5291 ± 0.0034	0.5377 ± 0.0066	0.3631 ± 0.0035
PrefKNN	0.3462 ± 0.0073	0.4048 ± 0.0038	0.3430 ± 0.0072	0.2720 ± 0.0037	0.3651 ± 0.0065	0.4283 ± 0.0024	0.3620 ± 0.0063	0.2904 ± 0.0023
PrefNMF	0.5778 ± 0.0112	0.5680 ± 0.0041	0.5724 ± 0.0109	0.3992 ± 0.0033	0.5883 ± 0.0073	0.5732 ± 0.0028	0.5832 ± 0.0073	0.4019 ± 0.0032
PrefCRF	$\mathbf{0.6206 \pm 0.0076}$	$\mathbf{0.5856 \pm 0.0028}$	$\mathbf{0.6150 \pm 0.0073}$	$\mathbf{0.4195 \pm 0.0028}$	$\mathbf{0.6395 \pm 0.0064}$	$\mathbf{0.5990 \pm 0.0023}$	$\mathbf{0.6340 \pm 0.0062}$	$\mathbf{0.4294 \pm 0.0021}$
	Given 50				Given 60			
Algorithm	NDCG@1	NDCG@10	MAP@1	MAP@10	NDCG@1	NDCG@10	MAP@1	MAP@10
UserKNN	0.3831 ± 0.0063	0.4424 ± 0.0027	0.3803 ± 0.0060	0.3015 ± 0.0026	0.4035 ± 0.0090	0.4622 ± 0.0035	0.4002 ± 0.0085	0.3163 ± 0.0027
NMF	0.5430 ± 0.0083	0.5326 ± 0.0036	0.5390 ± 0.0082	0.3669 ± 0.0025	0.5547 ± 0.0109	0.5409 ± 0.0063	0.5504 ± 0.0113	0.3734 ± 0.0055
PrefKNN	0.3831 ± 0.0092	0.4483 ± 0.0030	0.3803 ± 0.0089	0.3070 ± 0.0022	0.3979 ± 0.0075	0.4689 ± 0.0039	0.3948 ± 0.0069	0.3223 ± 0.0033
PrefNMF	0.5873 ± 0.0096	0.5745 ± 0.0035	0.5830 ± 0.0098	0.4019 ± 0.0033	0.5854 ± 0.0145	0.5733 ± 0.0048	0.5808 ± 0.0142	0.4007 ± 0.0037
PrefCRF	$\mathbf{0.6548 \pm 0.0055}$	$\mathbf{0.6068 \pm 0.0018}$	$\mathbf{0.6499 \pm 0.0059}$	$\mathbf{0.4372 \pm 0.0024}$	$\mathbf{0.6677 \pm 0.0074}$	$\mathbf{0.6139 \pm 0.0018}$	$\mathbf{0.6625 \pm 0.0072}$	$\mathbf{0.4436 \pm 0.0016}$

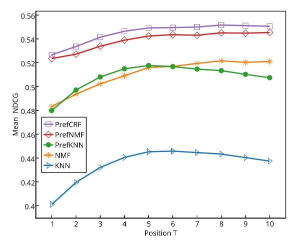
Fusing Preferences in Different Forms In this experiment, we first converted textual reviews into negative (-1), neutral (0), and positive (1) values using the *NLTK* library [2], and then converted them into *PR*. We study how these additional information can assist *PR*-based methods, and results over ten runs are shown in Table 2. Surprisingly, the performance of all *PR*-based methods except *PrefCRF* have decreased by incorporating textual reviews. We suspect that this is due to the misclassification errors introduced by sentiment classification on text. However, in the next subsection we will see that an accurate conversion can actually improve the performance.



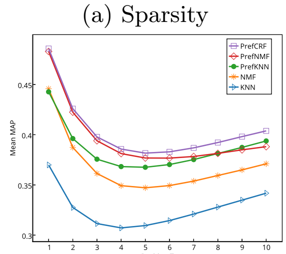
(a) Sparsity



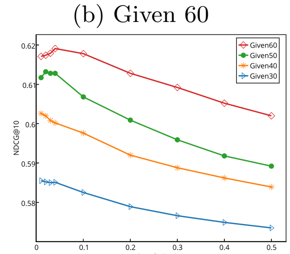
(b) Given 60



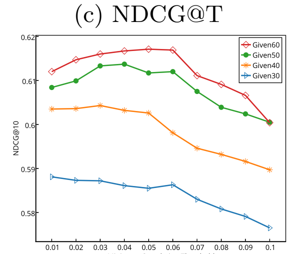
(c) NDCG@T



(d) MAP@T



(e) Regularization



(f) Parameters

Fig. 2: Plots of experimental results

Fusing Preferences in Different Scales In this experiment preferences in different scales are fused into *PR* to boost the performance of *PR*-based methods. The binary scale ratings are similar to the positive/negative textual reviews, however without incorrect values introduced by text classification.

Table 2: Results over ten runs on *Amazon* dataset.

	Ratings		Ratings + Textual Reviews	
Algorithm	NDCG@10	MAP@10	NDCG@10	MAP@10
UserKNN	0.6244 ± 0.0040	0.4599 ± 0.0035	0.6244 ± 0.0037	0.4599 ± 0.0025
NMF	$\mathbf{0.8073 \pm 0.0040}$	$\mathbf{0.6689 \pm 0.0038}$	$\mathbf{0.8073 \pm 0.0041}$	$\mathbf{0.6689 \pm 0.0000}$
PrefKNN	0.6410 ± 0.0038	0.4690 ± 0.0029	0.5765 ± 0.0039	0.4083 ± 0.0029
PrefNMF	0.7495 ± 0.0040	0.5924 ± 0.0031	0.7377 ± 0.0030	0.5806 ± 0.0031
PrefCRF	$\mathbf{0.8223 \pm 0.0033}$	$\mathbf{0.6813 \pm 0.0027}$	$\mathbf{0.8259 \pm 0.0035}$	$\mathbf{0.6890 \pm 0.0026}$

Table 3: Results over ten runs on *EachMovie* dataset.

	6-star Ratings		6-star Ratings + Binary Ratings	
Algorithm	NDCG@10	MAP@10	NDCG@10	MAP@10
UserKNN	0.4374 ± 0.0047	0.3418 ± 0.0029	0.4374 ± 0.0047	0.3418 ± 0.0029
NMF	0.5211 ± 0.0078	0.3710 ± 0.0034	0.5211 ± 0.0078	0.3710 ± 0.0034
PrefKNN	0.4908 ± 0.0070	0.3793 ± 0.0031	0.5074 ± 0.0061	0.3938 ± 0.0044
PrefNMF	0.5233 ± 0.0061	0.3820 ± 0.0033	0.5454 ± 0.0060	0.3881 ± 0.0032
PrefCRF	$\mathbf{0.5439 \pm 0.0056}$	$\mathbf{0.4006 \pm 0.0045}$	$\mathbf{0.5506 \pm 0.0053}$	$\mathbf{0.4038 \pm 0.0043}$

From Table 3, we can observe that the performance of all *PR*-based methods has increased by incorporating additional binary ratings, while the performance of rating-based methods remains the same.

Fusing Preferences with Different Biases In this experiment we investigate the impact of different biases, particularly the user-wise and item-wise biases, which are sampled from $Laplace(0, b)$. From Table 4 we can see that the performance of rating-based methods has decreased while *PR*-based methods are unaffected by such biases.

Table 4: NDCG@10 on *MovieLens*-20M dataset.

Algorithm	Bias = None	User-bias = $Laplace(0, 2)$	Item-bias = $Laplace(0, 2)$
UserKNN	0.4465 ± 0.0033	0.3729 ± 0.0033	0.2914 ± 0.0017
NMF	0.4982 ± 0.0034	0.4566 ± 0.0032	0.3074 ± 0.0019
PrefKNN	$\mathbf{0.4683 \pm 0.0027}$	$\mathbf{0.4683 \pm 0.0027}$	0.3157 ± 0.0021
PrefNMF	$\mathbf{0.4950 \pm 0.0035}$	$\mathbf{0.4950 \pm 0.0035}$	0.3137 ± 0.0017
PrefCRF	$\mathbf{0.5288 \pm 0.0037}$	$\mathbf{0.5288 \pm 0.0037}$	0.3729 ± 0.0023

Impact of Regularization and Correlation Threshold The proposed *PrefCRF* method has two user specified parameters: the *regularization coefficient* and a *minimum correlation threshold* that controls the number of correlation features. For the regularization, we can see from Fig. 2e that the performance gets better when a small regularization penalty applies. In other words, *PrefCRF* can generalize reasonable well without too much regularization. For the correlation threshold, Fig. 2f shows that a smaller threshold results better performance by including more correlation features, however, at the cost of more training time and more training data.

5 Conclusions and Future Works

In this paper we talcked the learning from different preference structures problem by the *PrefCRF* model, which takes advantages of both the representational power of the *CRF* and the ease of modeling *PR* by the *PrefNMF*. Experiment results on four public datasets demonstrate that different preference structures have been properly handled by *PrefCRF*, and significantly improved Top-N recommendation performance has been achieved. For future work, the computation efficiency of *PR*-based methods can be further improved given that the number of *PR* is usually much larger than ratings. Parallelization is feasible as as each user has a separated set of *PR* that can be processed simultaneously.

Acknowledgment

This work was partially supported by the Guangxi Key Laboratory of Trusted Software (No. KX201528).

References

1. J. Besag. Spatial interaction and the statistical analysis of lattice systems. *Journal of the Royal Statistical Society*, 36(2):192–236, 1974.
2. S. Bird, E. Klein, and E. Loper. *Natural language processing with Python*. O'Reilly Media, Inc., 2009.
3. A. Brun, A. Hamad, O. Buffet, and A. Boyer. Towards preference relations in recommender systems. In *PL: ECML/PKDD*, 2010.
4. O. Chapelle, D. Metlzer, Y. Zhang, and P. Grinspan. Expected reciprocal rank for graded relevance. In *CIKM'09*, pages 621–630. ACM, 2009.
5. M. S. Desarkar, R. Saxena, and S. Sarkar. Preference relation based matrix factorization for recommender systems. In *UMAP'12*, pages 63–75. Springer, 2012.
6. K. Järvelin and J. Kekäläinen. Cumulated gain-based evaluation of ir techniques. *ACM TOIS*, 20(4):422–446, 2002.
7. Y. Koren, R. Bell, and C. Volinsky. Matrix factorization techniques for recommender systems. *IEEE Computer*, 42(8):30–37, 2009.
8. Juntao Liu, Caihua Wu, Yi Xiong, and Wenyu Liu. List-wise probabilistic matrix factorization for recommendation. *Information Sciences*, 278:434–447, 2014.
9. Shaowu Liu, Gang Li, Truyen Tran, and Yuan Jiang. Preference relation-based markov random fields for recommender systems. *Machine Learning*, 106(4):523–546, 2017.
10. P. McCullagh. Regression models for ordinal data. *Journal of the Royal Statistical Society, Series B*, 42(2):109–142, 1980.
11. S. Rendle, C. Freudenthaler, Z. Gantner, and L. Schmidt-Thieme. Bpr: Bayesian personalized ranking from implicit feedback. In *UAI'09*, pages 452–461. AUAI Press, 2009.
12. P. Resnick, N. Iacovou, M. Suchak, P. Bergstrom, and J. Riedl. Grouplens: an open architecture for collaborative filtering of netnews. In *CSCW'94*, pages 175–186. ACM, 1994.
13. Daniel L Schacter and Chad S Dodson. Misattribution, false recognition and the sins of memory. *Philosophical Transactions of the Royal Society of London B: Biological Sciences*, 356(1413):1385–1393, 2001.
14. T. Tran, D. Q. Phung, and S. Venkatesh. Preference networks: Probabilistic models for recommendation systems. In *AusDM'07*, pages 195–202. ACS, 2007.