The Thirty-Third International
FLAIRS Conference (FLAIRS-33)

Modeling User Preferences Using Relative Feedback for Personalized Recommendations

Saikishore Kalloori
Media Technology Center
ETH Zurich, Switzerland
ssaikishore@ethz.ch

Tianyu Li
Rakuten Institute of Technology
Tokyo, Japan
tianyu.li@rakuten.com

Abstract

Recommender systems are widely developed to learn user preferences from their past history and make predictions on the unseen items a user may like. User preferences in the form of absolute preferences, such as user ratings or clicks are commonly used to model a user's interest and generate recommendations. However, rating items is not the most natural mechanism that users use for making decisions in daily life. For instance, we do not rate t-shirts when we want to buy one. It is more likely that we will compare them one to one, and purchase the preferred one. In this work, we focus on relative feedback, which generates pairwise preferences as an alternative way to model user preferences and compute recommendations. In our scenario, each user is shown a set of item pairs and asked to compare them to indicate which item in the pair is more preferred. We propose a recommendation algorithm to predict a user's relative preference for a given pairs of items and compute a personalised ranking of items. We demonstrate the effectiveness of our proposed algorithm in comparison with state-of-the-art relative feedback based recommendation approaches. Our experimental results reveal that the proposed algorithm is able to outperform the baseline algorithms on popular ranking-oriented evaluation metrics.

Introduction

Most research and applications of recommender systems (RSs) compute recommendations by exploiting user's preferences given items in the form of explicit or implicit feedback. For instance, Collaborative Filtering (CF) relies on a user-item rating matrix and generates recommendations by leveraging similarities between users or items based on available ratings (Ricci, Rokach, and Shapira 2015).

However, ratings have a few disadvantages associated to the fact that ratings are absolute evaluations. In addition, since ratings must be expressed in such a predefined rating scale (Gena et al. 2011), it creates some problems. For instance, a user who prefers one item over another one might end up giving the same rating to both items due to the limited rating scale. It could happen if a user likes an item and has already assigned the highest rating to the item, and then later wants to rate a second item that he prefers over the first one, has no choice but also give the same highest rating to the second item. Moreover, if most of items rated by a user contains 5 stars (on a one to five stars rating scale), then it is difficult to understand which items the user likes the most among them.

Recently, a few research works have been focusing on relative feedback as an alternative way of modeling user preferences to compute recommendations (Kalloori, Ricci, and Tkalcic 2016; Blédaité and Ricci 2015; Kalloori and Ricci 2017). There are also research that try to jointly model relative and absolute feedback data (Kalloori, Li, and Ricci 2019). However, in this work, we solely focus on relative feedback as an alternative to classical ratings (absolute feedback) and consider scenarios where users compare items in pairs, indicating which one, and to what extent, is preferred. When comparing two items, a numeric *pair score* is defined and this score indicates to what extent the first item is preferred over the second item (positive score), or if they are equivalent (null score) or if the second item is preferred (negative score). There are two types of relative feedbacks: implicit relative feedback and explicit relative feedback. In RSs, implicit relative feedbacks has been widely exploited using implicit data, e.g.: "clicked items are preferred to not clicked ones". However, in implicit relative feedbacks, which item among two clicked items should be preferred is not considered. In this work we focus on explicit relative feedback where the system presents users with pairs of items to compare and for any given pairs of items, the system predicts which item is preferred in the pair.

When RSs use relative feedbacks instead of ratings, the goal is : a) to predict unknown pair scores, since the users will score only a small subset of all the possible pairs of items, and b) to aggregate the available and predicted pair scores to produce personalized rankings of the items. In this work, we present a recommendation algorithm for predicting a user's relative preference for any given pairs of items. Our experiment setup very closely relate to real life recommendation scenario where we have a small training user data and a large number of test items to predict and rank. We conducted our experiments in comparison with state-of-the-art relative feedback based recommendation approaches and our results show that the proposed algorithm is able to predict unknown relative feedback preferences better than baseline algorithms on popular ranking-oriented evaluation metrics.

The rest of this paper is as follows. In the next section we illustrate the proposed ranking technique and this is followed by the description of the evaluation strategy used in our experiments and a comprehensive discussion of the obtained results. Finally, we formulate our conclusions and discuss future work.

The Proposed Method

Let U be the set of users and I be the set of items. Each user is described by a set of preferences over items in the form of relative feedback. We denote with P the set of relative preferences (comparison pairs). We denote the user u's relative feedback on the item pair (i, j) with r_{uij} and with $\hat{r}_{uij}$ the predicted relative feedback. The possible values for r_{uij} are:

$$r_{uij} = \begin{cases} 1 & \text{if user } u \text{ prefers } i \text{ over } j, \\ 0.5 & \text{if } i \text{ and } j \text{ are equally preferable to } u, \\ 0 & \text{if user } u \text{ prefers } j \text{ over } i. \end{cases} \tag{1}$$

Previous research emphasized the fact that having good rating prediction does not always translate into a better ranking of items (Cremonesi, Koren, and Turrin 2010). In this paper, we focus on ranking rather than rating prediction, and propose a ranking model called Collaborative Pairwise Ranking (CPR) to model relative feedback data. Since our training data contains relative feedback data as shown in equation 1, for a given a pair of items (i, j), we would like to predict user u preference order of that item pair, i.e., whether u prefers item i to j or item j to i. To predict the relative preference of the user u for pair (i, j), we have:

$$\hat{r}_{uij} = b_i - b_j + p_u^T q_i - p_u^T q_j, \tag{2}$$

where b_i denote the item i bias and p_u, q_i are d-dimensional latent factor vectors associated to user u and item i respectively. To find optimal parameters b_i, p_u and q_i, we use the following objective function:

$$\mathcal{L}_{pair}(\theta) = \min_{\theta} \sum_{r_{uij} \in P} (y(\hat{r}_{uij}) - r_{uij})^2 + \mathcal{R}(\theta). \tag{3}$$

$\mathcal{R}(\theta)$ is the regularizing term and θ are the model parameters b_i, p_u and q_i to be learned. In equation 3, we used $y(\hat{r}_{uij})$ which is defined using the following function:

$$y(\hat{r}_{uij}) = -r_{uij} ln(\sigma(\hat{r}_{uij})) - (1 - r_{uij}) ln(1 - \sigma(\hat{r}_{uij})), \tag{4}$$

where $\sigma(\hat{r}_{uij}) = \dfrac{1}{1 + e^{-\hat{r}_{uij}}}$ is used to map the predicted values between 0 and 1. We note that $y(r_{uij})$ is the binary cross entropy loss, which is also called log loss, and measures the prediction error (Kalloori, Li, and Ricci 2019). By utilizing the binary cross entropy we can view our ranking based recommendation as a binary classification problem (correct or incorrect preference order between item pairs). In this paper, we defined the above loss function for a implicit type of feedback but it can be easily extended to graded feedback such as five star ratings (Xue et al. 2017).

The objective function in equation 3 consists of pairwise loss function defined to model users relative feedback and in our experiments we learn all the model parameters by using stochastic gradient descent (SGD) algorithm by minimizing the (regularized) model's prediction error (on a training set of relative preference scores) (Koren, Bell, and Volinsky 2009). The parameter θ are updated as follows:

$$\theta \leftarrow \theta - \eta \left((y(\hat{r}_{uij}) - r_{uij}) * (\sigma(r_{uij}) - r_{uij}) * \frac{\partial \hat{r}_{uij}}{\partial \theta} + \lambda \theta \right), \tag{5}$$

where η is the learning rate and λ is the regularization coefficient. We also note that the proposed function in equation 4 has not been previously used for ranking in RSs as most of the RSs ranking methods, for instance, the BPR model (Rendle et al. 2009), use a sigmoid function. In this work, we explore it and show its effectiveness in our experiments. We note that once the parameter θ are learned, we predict the missing relative preference and aggregate them to compute a personalized item score ν_{ui} by averaging the $\hat{r}_{uij}$ predictions as follows:

$$\nu_{ui} = \frac{\sum_{j \in I \setminus \{i\}} \hat{r}_{uij}}{|I| - 1}. \tag{6}$$

For each user u a personalized ranking of items can be recommended by sorting items according to the ν_{ui} scores.

Experiments

Experimental Setup

This section describes the datasets, the evaluation procedure and an offline test results of the quality of the ranking list generated by the proposed method. To measure the performance of the proposed CPR model, we used three publicly available real-world datasets. The first one is MovieLens 100K which contains items for movies with $1 - 5$ stars ratings. We derive relative feedback by using two ratings of a same user and consider user prefers item i to item j if the rating of item i is higher than item j otherwise the opposite. The second dataset is Yahoo Music dataset which also consists of 1 to 5 stars ratings from 2400 users on 1000 songs and we derive relative feedback using the same procedure applied for MovieLens. Our third dataset contains relative feedback given by users participated in an online experiments (Blédaité and Ricci 2015), where authors developed a full movie recommender system based on relative feedback. The dataset contains 100 movies from MovieLens with 46 users participated in the experiment and a total of 2622 relative feedback data were collected. In addition to relative feedback data, the data set contains the ratings (converted to relative feedback) present in MovieLens-1m data for the 100 movies that were considered. We call this data set as MPAIR.

In our experiments, for data split we adopted the 'weak' generalization setting which has been already used in the literature (Balakrishnan and Chopra 2012; Volkovs and Zemel

Table 1: Recommendation performance for CPR and compared baseline algorithms on MPAIR dataset under different amount of preference present in the user profile. Significant results ($p < 0.05$) are marked with boldface

	Given N = 5			Given N = 10		
	Rank Hit	ppref@5	ppref@10	Rank Hit	ppref@5	ppref@10
CPR	0.520	**0.064**	**0.073**	0.545	**0.077**	**0.079**
MFP	0.490	0.046	0.049	0.491	0.052	0.054
NN-GK	0.455	0.024	0.030	0.486	0.022	0.028
NN-EDRC	0.480	0.028	0.032	0.485	0.029	0.033

Table 2: Recommendation performance for CPR and compared rating based algorithms on MovieLens dataset under different amount of preference present in the user profile. Significant results ($p < 0.05$) are marked with boldface

	Given N = 5			Given N = 10		
	MAP@10	MRR	Recall@10	MAP@10	MRR	Recall@10
CPR	**0.030**	**0.431**	0.064	**0.0140**	0.273	0.0334
MF-R	0.022	0.364	0.052	0.0108	0.248	0.0297
NN-PC	0.017	0.306	0.044	0.0068	0.228	0.0166

Table 3: Recommendation performance for CPR and compared rating based algorithms on Yahoo Music Dataset under different amount of preference present in the user profile. Significant results ($p < 0.05$) are marked with boldface

	Given N = 5			Given N = 10		
	MAP@10	MRR	Recall@10	MAP@10	MRR	Recall@10
CPR	**0.0137**	**0.171**	0.037	0.0110	0.122	**0.0370**
MF-R	0.0100	0.120	0.029	0.0098	0.108	0.0270
NN-PC	0.0089	0.114	0.027	0.0028	0.051	0.0097

2012). We first fix the number of training data per user profile $N = 5, 10$ and then randomly choose N preferences for each user for training and test on all the remaining preferences of the user. Such experimental setting allows the study of the ranking algorithms' sensitivity to the number of available training preferences per user. We note that the number of test items vary significantly across users, with many users having many more test data than training ones, thus simulating the real life recommendation scenario. We repeated the complete procedure five times and reported average performance.

For each test user we calculated the personalized ranked list using equation 6 and tested its quality by using three widely-adopted ranking metrics:

MRR: Mean Reciprocal Rank averages, for each user, the rank position in the recommended list of the test item appearing in the highest position.

Recall: Recall is defined as the ratio of the number of test items retrieved over the total number of items in the test set.

MAP: Mean Average Precision is the average of precision values at the rank positions where items present in test set occur. This is further averaged over all test users to give the final precision value.

Moreover, if the test set contains only relative feedback (MPAIR dataset contains only relative feedback in the test set), popular ranking metrics such as MAP cannot be used and we therefore use the following metrics:

Rank Hit: It measures the error between a set of relative preferences present in the test set and a ranked list. If for a pair (i, j) the user u prefers the item i over item j in the test set, then if the RS has ranked the item i above item j then we considered as a hit. We then define the Rank Hit as the total number of hits divided by the total number of relative preferences in the test set.

Precision of Preferences (**ppref@k**): This measure is a rank accuracy metric which evaluates a ranked list at a given cut-off rank k. The details of the metric can be found in (Carterette and Bennett 2008).

We consider user rating with 4 or 5 as relevant to a user and measure the ranking performance using MAP, MRR and Recall on MovieLens and Yahoo datasets. We have compared the proposed ranking model to two types of baseline algorithms. The first type is relative preferences based prediction models (Kalloori, Ricci, and Tkalcic 2016): (a) matrix factorization pair score prediction and item ranking methods (MFP) which extends Matrix Factorization for ratings data sets to MF for relative preference predictions (b) a user based Nearest-Neighbor (NN) approach for predicting unknown relative preferences that use two user-to-user similarity metrics. The first similarity metric is Goodman and Kruskal's gamma (GK) (referred as NN-GK) and the second similarity metric is Expected Discounted Rank Correlation (EDRC) (referred as NN-EDRC). The second type of baseline includes state-of-the-art rating based prediction algorithms: (a) Matrix Factorization (Koren, Bell, and Volinsky 2009) for rating prediction (referred as MF-R) (b) a NN approach for rating prediction and uses Pearson Similarity (Sarwar et al. 2001) based only on ratings (referred as NN-PC). Note that parameter settings for CPR, MFP and MF-R were carefully tuned and obtained used Nelder-Mead optimization method.

Evaluation Results

In our first experiment, we aim to understand the performance of CPR with relative preference based state-of-the-art algorithms and we compared the performance of CPR with MFP, NN-GK and NN-EDRC. Table 1 shows the ranking performance of CPR and the baseline approaches: for all the metrics, higher values denote better performance. We can observe that the proposed model, CPR, has better performance than the baseline models; it has better ranking accuracy across all the metrics. Our experimental analysis reveal that we are able to improve the relative feedback based state of the art prediction algorithms performance.

In our second experiment, we wanted to understand if predicting a user rating or relative preference is useful when computing a ranking of items. Therefore, we investigated if relative preference prediction algorithms generate better ranking of items than state-of-the-art rating prediction algorithms. Table 2 and table 3 show the recommendation performance of CPR with baseline rating prediction algorithms. Our proposed CPR generates better ranking performance compared to MF and NN-PC across all the metrics. We observe that the relative preference prediction is able to compute better recommendation than models performing rating predictions.

Conclusion and Future Work

In this paper, we proposed a ranking model that exploits relative feedback data. We presented a loss function that models relative feedback data and compute recommendations. Our experiment results show that the proposed model has a better ranking accuracy compared to state-of-the-art algorithms and also show that relative feedback can also used to model user preferences and to effectively build RSs.

Explicit relative feedback based modeling is a relatively new research compared to ratings. Our future work is to focus on active learning strategies for elicitation of relative preferences from users. Active learning for relative feedback elicitation is a necessary component when one needs to build relative preferences based RSs. We also want to better investigate how to combine both relative and rating preferences and to develop mixed active learning strategies (Kalloori, Ricci, and Gennari 2018) that can propose to the users specific items to rate and item pairs to compare.

Acknowledgement

This work was partially supported by Tamedia, Ringier, NZZ, SRG, VSM, viscom, and the ETH Zurich Foundation.

References

Balakrishnan, S., and Chopra, S. 2012. Collaborative ranking. In *Proceedings of the fifth ACM international conference on Web search and data mining*, 143–152. ACM.

Blédaité, L., and Ricci, F. 2015. Pairwise preferences elicitation and exploitation for conversational collaborative filtering. In *Proceedings of the 26th ACM Conference on Hypertext & Social Media*, 231–236. ACM.

Carterette, B., and Bennett, P. N. 2008. Evaluation measures for preference judgments. In *Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*, 685–686. ACM.

Cremonesi, P.; Koren, Y.; and Turrin, R. 2010. Performance of recommender algorithms on top-n recommendation tasks. In *Proceedings of the fourth ACM conference on Recommender systems*, 39–46. ACM.

Gena, C.; Brogi, R.; Cena, F.; and Vernero, F. 2011. The impact of rating scales on user's rating behavior. In *International Conference on User Modeling, Adaptation, and Personalization*, 123–134. Springer.

Kalloori, S., and Ricci, F. 2017. Improving cold start recommendation by mapping feature-based preferences to item comparisons. In *Proceedings of the 25th Conference on User Modeling, Adaptation and Personalization*. ACM.

Kalloori, S.; Li, T.; and Ricci, F. 2019. Item recommendation by combining relative and absolute feedback data. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, 933–936.

Kalloori, S.; Ricci, F.; and Gennari, R. 2018. Eliciting pairwise preferences in recommender systems. In *Proceedings of the 12th ACM Conference on Recommender Systems*, 329–337. ACM.

Kalloori, S.; Ricci, F.; and Tkalcic, M. 2016. Pairwise preferences based matrix factorization and nearest neighbor recommendation techniques. In *Proceedings of the 10th ACM Conference on Recommender Systems*, 143–146. ACM.

Koren, Y.; Bell, R.; and Volinsky, C. 2009. Matrix factorization techniques for recommender systems. *Computer* (8):30–37.

Rendle, S.; Freudenthaler, C.; Gantner, Z.; and Schmidt-Thieme, L. 2009. Bpr: Bayesian personalized ranking from implicit feedback. In *Proceedings of the twenty-fifth conference on uncertainty in artificial intelligence*. AUAI Press.

Ricci, F.; Rokach, L.; and Shapira, B. 2015. Recommender systems: Introduction and challenges. In *Recommender Systems Handbook*. Springer.

Sarwar, B. M.; Karypis, G.; Konstan, J. A.; Riedl, J.; et al. 2001. Item-based collaborative filtering recommendation algorithms. *Www* 1:285–295.

Volkovs, M., and Zemel, R. S. 2012. Collaborative ranking with 17 parameters. In *Advances in neural information processing systems*, 2294–2302.

Xue, H.-J.; Dai, X.; Zhang, J.; Huang, S.; and Chen, J. 2017. Deep matrix factorization models for recommender systems. In *IJCAI*, 3203–3209.