Pairwise Preferences Based Matrix Factorization and Nearest Neighbor Recommendation Techniques

Saikishore Kalloori, Francesco Ricci, and Marko Tkalcic
Faculty of Computer Science
Free University of Bozen - Bolzano
Piazza Domenicani 3, I - 39100, Bolzano
ksaikishore@unibz.it, fricci@unibz.it, marko.tkalcic@gmail.com

ABSTRACT

Many recommendation techniques rely on the knowledge of preferences data in the form of ratings for items. In this paper, we focus on pairwise preferences as an alternative way for acquiring user preferences and building recommendations. In our scenario, users provide pairwise preference scores for a set of item pairs, indicating how much one item in each pair is preferred to the other. We propose a matrix factorization (MF) and a nearest neighbor (NN) prediction techniques for pairwise preference scores. Our MF solution maps users and items pairs to a joint latent features vector space, while the proposed NN algorithm leverages specific user-to-user similarity functions well suited for comparing users preferences of that type. We compare our approaches to state of the art solutions and show that our solutions produce more accurate pairwise preferences and ranking predictions.

Keywords

Pairwise Preferences; Matrix Factorization; User Similarity

1. INTRODUCTION

Recommender Systems (RSs) address the information overload problem by providing users with personalized suggestions for items that are likely to be interesting and relevant to their needs. Many RSs are rating-based, i.e., user preferences for items are expressed in the form of absolute evaluations in a predefined scale, e.g., the classical five stars scale that is used in Amazon.com.

However, rating-based RSs have limitations related to the fact that ratings are absolute evaluations with respect to a benchmark. For instance, a user who prefers one item over another might end up giving the same rating to both of them due to the limited rating scale. Also, consider a user who has liked an item and has assigned the highest rating to this item. If this user subsequently finds a new item that she prefers over the first one, she has no choice but to also give it the highest rating [3]. Considering these limitations some authors have developed recommendation techniques that leverage an alternative way to express user preferences, namely, pairwise preferences [3, 6, 2]. In these approaches users can express their preference for one of the items in a pair (i, j) with a pair score, i.e., a positive or negative number. The larger positive (smaller negative) the score is the more the item i (j) is preferred to the item j (i). When pairwise preference scores are used instead of ratings, the core problem for the RS 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 [2] it is shown that it is possible to build a RS using pairwise preference scores that performs better than state of the art rating-only based solutions (in terms of nDCG and precision). The Personalized Differential Matrix with Ratings and Pairwise Preferences (PPR) method was proposed there. It can use a collection of pre-existing ratings and pairwise preference scores to produce a personalized ranking for each user. PPR is a nearest neighbor (NN) approach, and uses a generalization of Pearson correlation as user-to-user similarity for user profiles composed of pair scores. In this paper we conjecture that the performance of the PPR can be improved in two ways. Firstly, we believe that the user-to-user similarity metric used in PPR is not appropriate for pair scores based profiles, hence we conjecture that a better suited similarity metric can improve the items ranking accuracy. Secondly, when the data is sparse, Matrix Factorization (MF) has been shown to give even better results in many domains [9]. Hence, we conjecture that MF can also improve pair scores prediction and we propose a MF technique that finds latent feature models for pairs of items, similarly to how in standard MF single items are modeled by latent features. We carried out an offline experiment to compare our approaches using a movie data set that contains pairwise preference scores given to movie pairs and we show that the proposed approaches substantially improve state of the art ranking algorithms.

2. RELATED WORK

In the survey conducted in [8] it emerges that users prefer providing preferences in the form of pairwise scores and [5] shows that making pairwise preference judgments are faster than absolute judgments (ratings). In [2], it has also been shown that pairwise preferences can be as easy to enter as ratings when a suitable graphical user interface is provided.

RecSys '16, September 15 - 19, 2016, Boston , MA, USA

ISBN 978-1-4503-4035-9/16/09…$15.00
DOI: http://dx.doi.org/10.1145/2959100.2959142

And in [8], by conducting a user study, they found that item comparisons are 20% more stable over the time. Moreover in [2] it has shown that by using pairwise preferences one can produce better recommendations than ratings by using pairwise preferences almost equal to number of ratings.

In [3], an algorithm that uses pairwise preferences for predicting item ratings was proposed. It implements a NN collaborative filtering approach where user-to-user similarity is measured as the number of item pairs on which both users have the same preferences. They used this special type of similarity for predicting ratings (not pairwise scores) and concluded that exploiting preference relations does not lead to a decrease in the quality of rating predictions. In [1], the authors proposed a novel method to evaluate the rank accuracy of a RS assuming that the user preferences are expressed as pairwise preferences. In this paper we show that this evaluation method can in fact be used as user-to-user similarity in a pair scores prediction algorithm.

In [6] a MF technique for pairwise preferences that is similar to ours was proposed. They represent each item using a d-dimensional vector and estimate pairwise preference scores using an inverse-logit function. Quite differently, in our matrix factorization technique we consider a d-dimensional vector representation for each item pair (i, j) and we predict pairwise scores as in classical MF one predict ratings. BPR (Bayesian Personalised Ranking) [10] is a popular item ranking techniques that was applied to pairwise preference data implicitly derived from users' online clicks.

3. PAIR SCORES PREDICTION

We present two novel methods for pair scores prediction. The first extends MF for ratings data sets [9] to MF for pair scores predictions. The second is a NN approach and we introduce two new user-to-user similarity measures suited to assess the similarity of two profiles based on pair scores.

Let U be a set of users and I be a set of items. We denote with r_{uij} the pair score given by user u to the pair of items (i, j), and r^*_{uij} a predicted pair score. In the data set that we have used the possible pair score values are $[-4, 4]$, since they were acquired in an experiment where a pair scores based recommender was compared with a 5 stars rating based one. In a rating-based RS, a user $u \in U$ expresses her preferences for items with ratings r_{ui}. If the ratings r_{ui} and r_{uj} are present then one can form the pair score r_{uij} as $r_{ui} - r_{uj}$.

3.1 MF for Pair Scores Prediction

We present here a matrix factorization pair score prediction and item ranking method (MFP). MFP models each user and item pair as d-dimensional vectors p_u and q_{ij} respectively. Let $\mathbf{R}$ be the matrix that contains all the pair scores that the users have assigned. The element r_{uij} in the matrix (tensor) is the pair score the user u gave to the item pair (i, j). MFP factorizes $\mathbf{R}$ into two latent matrices of lower dimension d: the user-factor matrix $\mathbf{P}$ and the item-pair-factor matrix $\mathbf{Q}$. The prediction for a missing pair score of the user u for a pair (i, j) is:

$$r^*_{uij} = \mu + b_u + b_{ij} + p_u^T q_{ij}$$

where μ is the overall average of all the pair scores, and b_u and b_{ij} are the baseline parameters that model the observed deviations from the average pair score due to the user u and item pair (i, j), respectively. In our experiments we learn all the model parameters by using stochastic gradient descent [9], by minimizing the (regularized) model's prediction error (on a training set of pair scores) as $r_{uij} - r^*_{uij}$. As a result of the learning process we get a full matrix of predicted pair scores. We can then compute a personalized item score ν_{ui} by averaging the r^*_{uij} predictions, as done by [2]:

$$\nu_{ui} = \frac{\sum_{j \in I \setminus \{i\}} r^*_{uij}}{|I|} \tag{1}$$

For each user u the items are then ranked by descending values of ν_{ui}.

3.2 NN for Pair Scores Prediction

In previous research a NN approach was used for pair scores data by using Pearson similarity to calculate user-to-user similarities and to generate predictions [2]. In the following example we will illustrate why Pearson similarity is not suitable in this case, and we propose alternative similarity measures that better fit to pairwise scores profiles. **Example 1:** Looking at the data shown in Table 1, one can see that u and v should not be considered similar because u prefers k over j and l over k whereas v prefers j over k and k over l. However, Pearson similarity yields 0.51, which is a relative high value.

Table 1: Pair scores of two users for items $\{i, j, k, l\}$

	(i,j)	(i,k)	(i,l)	(j,k)	(j,l)	(k,l)
u	-	1	-	-1	-	-2
v	-	-	-	3	-	1

We now present two user-to-user similarity measures that let the recommender to better predict missing pair scores: (i) Goodman and Kruskal's gamma (GK) and (ii) Expected Discounted Rank Correlation (EDRC). Let $v \in U_{ij}$ be the set of all users for which we have the pair score r_{vij}. The NN prediction formula for missing pairwise scores is:

$$r^*_{uij} = \frac{1}{\sum_{v \in U_{ij}} sim(u,v)} \sum_{v \in U_{ij}} sim(u,v) * r_{vij} \tag{2}$$

where $sim(u, v)$ is either EDRC, GK, or Pearson Correlation, as in [2]. After having predicted the missing pair scores, item scores and ranking can be computed again using Equation 1.

3.2.1 Goodman and Kruskal's gamma

Goodman and Kruskal's gamma (GK) was proposed in [7]. It is a symmetric measure of correlation. Given two user profiles u and v with their pair scores, $GK(u, v)$ is calculated using two quantities: (a) the number P of item pairs that are ranked in the same order in both profiles (concordant pairs) and (b) the number Q of item pairs that are ranked in the reversed order (reversed pairs):

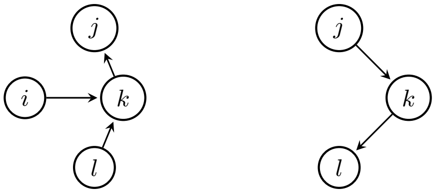
$$GK(u,v) = \frac{P - Q}{P + Q} \tag{3}$$

GK ranges from -1 (100% negative association) to +1 (100% positive association). In Example 1, using GK, we obtain a more meaningful similarity value $GK(u, v) = -1$.

3.2.2 Expected Discounted Rank Correlation

In [1], the authors proposed a novel method, *expected discounted rank correlation* (EDRC) to evaluate rank accuracy of a recommendation list. EDRC measures the similarity between two sets of pairwise preferences: (i) a ground truth set of pairwise preferences and (ii) a set of predicted pairwise preferences. In their method, the authors require to have the same items in both sets (although not necessarily the same comparisons of items). But in our scenario we measure the similarity of two user that may have compared different items. Thus, we take the union of items appearing in the pairs compared by the two users and we change the way EDRC deals with missing pairwise preferences (explained later).

EDRC is asymmetric and measures how much a user u pairwise preferences match a user v pairwise preferences. It is based on a graph representation of the pair scores where the vertices represent items and edges represent pairwise preferences (See Figure 1). Hence, EDRC, as well as GK, ignores the magnitude of the pair scores and considers only their sign.



(a) User u (b) User v

Figure 1: Users preference graphs from Table 1 data

In order to calculate EDRC between two users u and v, a rank score $R(\cdot)$, a discount weight $D(\cdot)$ and a score $C(\cdot)$ for each item present in u profile is required. The rank score $R(\cdot)$ is computed by Algorithm 1, and we assign a linear discount weight $D(\cdot) = R(\cdot)$. Algorithm 1 is a modified version of the algorithm presented in [1] and is not applicable if the preference graph of u contains cycles. If a cycle is present, then we must first collapse together the vertices in each cycle and treat them as one vertex. The rank value is then the same for all vertices in the cycle[1]. To calculate the score $C(\cdot)$, we take the union of the items present in both users profiles and we call it I. For example, in Figure 1, we have $I = \{i, j, k, l\}$. Using pairwise preferences of both users u and v, we compute the score $C(\cdot)$ for each item present in the profile of user u. The score $C(\cdot)$ of an item represents the overall preferences agreement of the two users between an item and other items present in I. For instance, for the item $j \in I$, the score $C(j)$ is the sum of the agreements that the two users have on the pairs (j, i), (j, k), (j, l). We used the same approach proposed in [1] to calculate the $C(\cdot)$ score, and the reader is referred to that reference for further details. However, our scenario is different because there exist cases where the relationship between (i, j), i.e, which item is preferred, is unknown. In that case we assume that there is 50% likelihood for i to be preferred to j and 50% likelihood for the opposite preference. This situation can be seen in the example shown in Figure 1, where preference relationship between the items present in v profile and item i are unknown. We finally calculate the similarity of two users as follows:

Algorithm 1 Algorithm for Calculating Rank Scores

```
$OUT(i, G)$ is set of outgoing vertices from $i$
procedure RankScore($G(V, E)$)
    for each vertex $i \in V$ do
        $R(i) \leftarrow 1$
    end for
    $L \leftarrow TopologicalSort(G)$    ▷ topological order of vertices
    for each vertex $i \in L$ do
        for all $j \in OUT(i, G)$ do
            $R(j) \leftarrow MAX(R(j), R(i) + 1)$
        end for
    end for
end procedure
```

$$EDRC(u, v) = \frac{2}{Z} \cdot \left[\sum_{i \in S} \frac{C(i)}{D(i)} \right] - 1 \tag{4}$$

where S is a set of vertices having an incoming edge in u and Z is a normalization factor to ensure that the value of this similarity stays between -1 and +1. Referring again to Example 1, one can show that $EDRC(u, v) = -0.41$, hence again a negative value, which is appropriate in this case.

4. EVALUATION PROCEDURE

In order to study the performance of MFP, and the adoption of the GK and EDRC as user-to-user similarities in a NN method (referred to as NN-GK and NN-EDRC, respectively) we compare them with three state-of-the-art algorithms: (i) NN using Pearson correlation as user-to-user similarity (PPR) [2] (ii) Matrix Factorization approach using preference relations (MFPref) [6] and (iii) BPR-MF [10].

The dataset[2] used in our experiment was acquired through an online experiment described in [2]. The authors developed a full movie recommender system based on pairwise preferences, which included a preference elicitation interface to show items pairs to users and to collect their pair scores. They used 100 movies from Movielens, 46 users participated in the experiment and a total of 2622 pairwise preferences were collected. In addition to pair scores, the data set contains the ratings present in MovieLens 100K data for the 100 movies that were considered. There were a total of 73078 ratings entered by 1128 users and we converted them to pair scores. The sparsity of our data set is 99.8%.

In our experiments we performed a five-fold cross validation. The pair scores generated from the MovieLens ratings were part of the training set. We shuffled the pairwise preferences of the 46 users and split them into five (roughly) equally sized subsets. Then, for each iteration, the system prediction model was trained using four of these five subsets and the testing was done on the fifth subset. Since our ground truth contains incomplete pairwise preferences and not full rankings of items, rank accuracy like NDCG and Average Precision of a recommender systems are inapplicable. Therefore, for each iteration, we used the following metrics in order to evaluate our approach:

- **Predict Hit:** Assume that a set of pair scores in the test set T is known and the system has predicted these pair scores:

$$PredictHit = \sum_{r_{uij} \in T} PH(r_{uij}) / |T| \tag{5}$$

$PH(r_{uij})$ is 1 if the sign of a pair score contained in the test set T for the item pair (i, j) is correctly predicted by the RS and 0 otherwise.

[1] However, we note that in our data we did not have cycles.

[2] Available at: http://www.inf.unibz.it/~kalloori/

- **Rank Hit:** It measures the ranking error between a set of pair scores present in the test set and a ranked list. For each user, his personalized ranked list is constructed using Equation 1.

$$RankHit = \sum_{r_{uij} \in T} RH(r_{uij})/|T| \qquad (6)$$

  where $RH(r_{uij})$ is 1 if the RS has ranked for u the item i above item j and the user u does prefer the item i over item j and 0 otherwise.
- **Precision of Preferences (ppref@k):** This measure is a rank accuracy metric which evaluates a ranked list at a given cut-off rank k. In [4] a pair (i, j) is defined as *ordered* by the RS if at least one item appears above rank k and unordered otherwise. A pair (i, j) is defined as *correctly ordered* for a user if the RS's ordering matches his preference order. *Precision of preferences at k (ppref@k)* is defined as the total number of correctly ordered pairs divided by the total number of ordered pairs [4].

5. RESULTS

The recommendation performances of the proposed algorithms[3] for the considered metrics are shown in Table 2. Higher values indicates better performance.

Table 2: Ranking and prediction accuracy of the compared algorithms. Best results are marked with boldface.

	Predict hit	Rank hit	ppref@5	ppref@10
NN-EDRC	**0.701** ±0.04	**0.791** ±0.01	**0.077** ±0.03	**0.048** ±0.01
MFP	0.585 ±0.06	0.733 ±0.01	0.053 ±0.01	0.027 ±0.00
NN-GK	0.407 ±0.06	0.624 ±0.04	0.033 ±0.01	0.024 ±0.00
MFPref	0.563 ±0.04	0.571 ±0.03	0.043 ±0.03	0.015 ±0.01
BPR-MF	0.501 ±0.05	0.515 ±0.08	0.026 ±0.06	0.010 ±0.02
PPR	0.257 ±0.03	0.281 ±0.03	0.061 ±0.00	0.021 ±0.00

As it can be noted, NN-EDRC predicts pair score signs and ranks items more accurately than the other methods. Observing Table 2 we can make the following conclusions.

The Nearest Neighbor based method EDRC is the best prediction algorithm that we have compared in the experiment. Our experiment results suggest that in order to assess the similarity between two users, when their preferences are expressed as pair scores, a specific similarity metric should be used, instead of relying on Pearson correlation, which is more appropriate for ratings data sets.

Comparing MFP with MFPref, we can conclude that our proposed matrix factorization approach, MFP, can better capture user and items pairs latent features and compute predictions than MFPref. This may depend on the fact that MFP takes into account the original preference scores (in the range from -4 to 4), while MFPref considers only the sign of the pair scores.

BPR-MF does not yield better results. It is worth noting that BPR-MF is not designed to predict pair scores but only a ranking compatible with the observed pair scores. We conjecture that this explains why it has, in this data set and application, an inferior performance compared to a method originally designed to predict pair scores.

6. CONCLUSIONS AND FUTURE WORK

In this paper we have introduced two techniques for computing recommendations by leveraging user preferences expressed with pairwise scores. In particular, we have proposed two user-user similarity metrics, to be used in nearest neighbor based pair scores prediction methods and, a straightforward extension to pairwise preference data of matrix factorization for ratings data sets. The empirical evaluation performed on a movie data set has shown that our approaches have better prediction accuracy compared to state of the art baselines in terms of Predict Hit, Rank Hit, ppref@5 and ppref@10.

We have performed our offline tests on a small dataset. Hence, in the future we will generate a larger data sets by integrating the proposed recommendation techniques in a real system. In fact one of the major limitation of our approach is the quadratic grow with the number of items of the pair scores matrix. However one must note that only a small minority of all the potential pairs of items will be actually compared by the users.

Furthermore, we plan to develop viable active learning strategies to identify the pairs that each user should compare. In the future, we will also conduct live user experiments and identify specific conditions and situations where pairwise preferences elicitation is meaningful and beneficial.

7. REFERENCES

[1] B. Ackerman and Y. Chen. Evaluating rank accuracy based on incomplete pairwise preferences. In *Proc. Workshop on UCERSTI Recsys '11*, 2011.

[2] L. Blédaité and F. Ricci. Pairwise preferences elicitation and exploitation for conversational collaborative filtering. In *Proc. Hypertext & Social Media '15*, pages 231–236, 2015.

[3] A. Brun, A. Hamad, O. Buffet, and A. Boyer. Towards preference relations in recommender systems. In *Preference Learning Workshop (ECML-PKDD '10)*, 2010.

[4] B. Carterette and P. N. Bennett. Evaluation measures for preference judgments. In *Proc. SIGIR '08*, pages 685–686. ACM, 2008.

[5] B. Carterette, P. N. Bennett, D. M. Chickering, and S. T. Dumais. Here or there: Preference judgments for relevance. In *Advances in Information Retrieval 2008*, pages 16–27. Springer, 2008.

[6] M. S. Desarkar, R. Saxena, and S. Sarkar. Preference relation based matrix factorization for recommender systems. In *UMAP '12*, pages 63–75. Springer, 2012.

[7] L. A. Goodman and W. H. Kruskal. *Measures of association for cross classifications.* Springer, 1979.

[8] N. Jones, A. Brun, and A. Boyer. Comparisons instead of ratings: Towards more stable preferences. In *Proc. WI-IAT '11*, pages 451–456, 2011.

[9] Y. Koren, R. Bell, and C. Volinsky. Matrix factorization techniques for recommender systems. *Computer*, 42(8):30–37, 2009.

[10] S. Rendle, C. Freudenthaler, Z. Gantner, and L. Schmidt-Thieme. Bpr: Bayesian personalized ranking from implicit feedback. In *Proc. UAI '09*, pages 452–461, 2009.

[3] MFP: d=15 and γ=0.003 and MFPref: d=15 and γ=0.008 and BPRMF: d=20 and γ=0.0025