Item Recommendation by Combining Relative and Absolute Feedback Data

Saikishore Kalloori*
Free University of Bozen-Bolzano, Italy
ksaikishore@unibz.it

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

Francesco Ricci
Free University of Bozen-Bolzano, Italy
fricci@unibz.it

ABSTRACT

User preferences in the form of absolute feedback, s.a., ratings, are widely exploited in Recommender Systems (RSs). Recent research has explored the usage of preferences expressed with pairwise comparisons, which signal relative feedback. It has been shown that pairwise comparisons can be effectively combined with ratings, but, it is important to fine tune the technique that leverages both types of feedback. Previous approaches train a single model by converting ratings into pairwise comparisons, and then use only that type of data. However, we claim that these two types of preferences reveal different information about users interests and should be exploited differently. Hence, in this work, we develop a ranking technique that separately exploits absolute and relative preferences in a hybrid model. In particular, we propose a joint loss function which is computed on both absolute and relative preferences of users. Our proposed ranking model uses pairwise comparisons data to predict the user's preference order between pairs of items and uses ratings to push high rated (relevant) items to the top of the ranking. Experimental results on three different data sets demonstrate that the proposed technique outperforms competitive baseline algorithms on popular ranking-oriented evaluation metrics.

CCS CONCEPTS

• **Information systems → Recommender systems**; **Personalization**.

KEYWORDS

Relative Feedback, Absolute Feedback, Pairwise Preferences, Ratings, Recommender System

ACM Reference Format:
Saikishore Kalloori, Tianyu Li, and Francesco Ricci. 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 (SIGIR '19), July 21–25, 2019, Paris, France.* ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/3331184.3331295

*This work was done when the author was intern with Rakuten

ACM ISBN 978-1-4503-6172-9/19/07…$15.00
https://doi.org/10.1145/3331184.3331295

1 INTRODUCTION

Recommender systems (RSs) exploit user's preferences for items either in the form of explicit feedback, such as ratings, or implicit feedback, such as clicks. Both of them are absolute preferences, which means that the evaluated item is not compared to another one. However, it has been shown that such absolute preferences have few disadvantages [2, 5]. For instance, if a user rates most of the items 5, then it is impossible to understand which one the user really prefers. Recently, more and more authors focused on exploiting relative preference data, such as pairwise comparisons, as an alternative way to model user preferences and compute recommendations [2–5]. In these scenarios users either explicitly compare item pairs and say which one is preferred or such comparisons are derived from absolute preferences, e.g.: "clicked items are preferred to not clicked ones".

Actually, in [3, 5], it was shown that pairwise comparisons (relative preferences) can be effectively incorporated in RSs by training matrix factorization and nearest neighbour approaches. In [4] it is presented a mobile RS and it is shown that, when the user is searching for a rather specific recommendation (e.g., a restaurant for a dinner with friends), by eliciting from the user pairwise comparisons, the system can offer a better recommendation experience compared to when is eliciting ratings. However, the previous approaches, which are based on both type of data, actually convert the available ratings into pairwise comparisons, by taking the difference of the ratings for two items, and train a single prediction model by using only pairwise comparisons [2–5]. Other approaches derive pairwise comparisons from item clicks, by assuming that a clicked item is preferred to a non-clicked one [6, 8]. But, we claim that these two types of preferences (ratings/clicks and pairwise comparisons) reveal different type of information about users' interests and should be exploited differently. Therefore, in order to better use both types of preference data it is necessary to fine-tune the adopted recommendation technique (ranking and preference learning) such that each type of data is optimally exploited.

In this paper, we focus on such situations where both pairwise comparisons and ratings are used. We propose a novel ranking approach, called Joint Pairwise comparisons and Ratings (JPR), that leverages both absolute (ratings) and relative (comparisons) preferences of users. Our developed ranking algorithm employs a joint loss function to model ratings and pairwise comparisons of users. The produced model predicts the preference order of item pairs by using pairwise comparisons data and exploits the available user ratings to push the user's highly rated (relevant) items to top of the ranking. We conducted offline experiments to evaluate the proposed ranking model and we compared it with state of the art ranking algorithms. Our experimental results demonstrate that this new ranking model has a better performance than state of the art baseline algorithms. Hence, our results show that the proposed learning procedure is effective to model both relative and absolute feedback data.

The rest of this paper is structured as follows. In the next section we illustrate, the proposed ranking technique for computing personalized ranking of items. 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.

2 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 and absolute feedback. We denote with P the set of relative preferences (comparison pairs) and R the set of absolute preferences (ratings). Let r_{ui} denotes the user u's absolute feedback on item i such as the classical 5-stars item ratings and $\hat{r}_{ui}$ the predicted absolute feedback. From the absolute feedback data, for each user u, we define I_u^+ the set of relevant items, for instance, those with ratings 4 and 5, I_u^- the set of irrelevant items, with ratings 1, 2 and 3, and $n_u = |I_u^+ \cup I_u^-|$ is the total number of his ratings. We also 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 $\{1, 0\}$, where $r_{uij} = 1$ implies user u prefers item i over item j and 0 implies the opposite.

In the following subsections, we describe our hybrid approach that jointly models both type of preference data. We will first describe how we exploit pairwise comparisons data and next how we exploit ratings data.

2.1 Collaborative Pairwise Ranking - CPR

In this paper, we are interested in ranking rather than rating prediction, therefore we focus on a ranking-oriented collaborative filtering [8] approach that aims at predicting a ranking of items. 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 pairwise comparison of the user u for pair (i, j), we use the following model [8]:

$$\hat{r}_{uij} = b_i - b_j + p_u^T q_i - p_u^T q_j \tag{1}$$

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 loss function:

$$y(\hat{r}_{uij}) = -r_{uij} ln(\sigma(\hat{r}_{uij})) - (1 - r_{uij}) ln(1 - \sigma(\hat{r}_{uij})) \tag{2}$$

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 [7]. By utilizing the binary cross entropy we can view our pairwise ranking based recommendation as a binary classification problem (correct or incorrect preference order between item pairs). With the above setting we define the following objective function:

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

$\mathcal{R}(\theta)$ is the regularizing term and θ are the model parameters to be learned. We follow the widely used stochastic gradient descent (SGD) algorithm to optimize the objective function and the parameter θ are updated as follows:

$$\theta \leftarrow \theta - \eta \left((\sigma(r_{uij}) - r_{uij}) * \frac{\partial \hat{r}_{uij}}{\partial \theta} + \lambda \theta \right) \tag{4}$$

where η is the learning rate and λ is the regularization coefficient. We note that in Eq 4 the learning rate is controlled by the term $\sigma(r_{uij}) - r_{uij}$. This means that if the model prediction error is large (small) then a larger (smaller) update is made. We also note that the proposed loss function has not been previously used for ranking in RSs as most of the RSs ranking methods, for instance, the BPR model [8], use a sigmoid function.

2.2 Push Relevant Items to Top - Push

We will now describe how we exploit absolute feedback data. Let us assume that each user u has expressed a set of absolute feedback on items and, as mentioned before, using this data, we can derive his relevant and irrelevant items. The goal here is to correctly rank these relevant items, i.e., to put items with the highest rating, on top of the ranked list and to push down from the top the irrelevant ones [1]. In order to do that, we define the "height" of a relevant item $i \in I_u^+$ as the number of irrelevant items ranked above it:

$$H_u(i) = \sum_{j \in I_u^-} \mathbb{1}[\hat{r}_{ui} < \hat{r}_{uj}] \tag{5}$$

where $\mathbb{1}[\cdot]$ is the indicator function. Absolute preferences are predicted as $\hat{r}_{ui} = b_i + p_u^T q_i$ and the same parameters θ used in CPR are also used here. We note that a high value of $H_u(i)$ implies that there are several irrelevant items ranked above the relevant item i. The goal is to make the height of all the relevant items as close to zero as possible, therefore, implicitly, to push the relevant items to the top. However, the indicator function is not suitable for optimization, since it cannot be differentiated, therefore, we use a surrogate for that function, that is given as follows:

$$\mathcal{H}_u(i) = \sum_{j \in I_u^-} log(1 + exp(-(\hat{r}_{ui} - \hat{r}_{uj}))) \tag{6}$$

Since our goal is to reduce the height of relevant items to zero, using the above setting, the objective function of the push model can be formulated as a minimization problem:

$$\mathcal{L}_{Push}(\theta) = \min_{\theta} \sum_{u \in U, i \in I_u^+} \frac{1}{n_u} (\mathcal{H}_u(i))^2 + \mathcal{R}(\theta) \tag{7}$$

We use again stochastic gradient descent (SGD) to optimize the objective function and the update rules for a user u and a relevant item i are as follows:

$$\theta \leftarrow \theta - \eta \left(\frac{2 * \mathcal{H}_u(i)}{n_u} * \left(\sum_{j \in I_u^-} \frac{-1}{1 + e^{(\hat{r}_{ui} - \hat{r}_{uj})}} \right) \frac{\partial(\hat{r}_{ui} - \hat{r}_{uj})}{\partial \theta} + \lambda\theta \right) \quad (8)$$

2.3 Joint Learning using Relative and Absolute Preferences

We note that the CPR model uses relative preferences data while the Push model uses absolute preferences on items. In order to build a RS that leverages both types of preferences, we have developed a specific leaning to rank method, which is called Joint Pairwise preferences and Ratings (JPR). JPR uses a novel joint loss function combining CPR and Push to perform joint learning. More specifically, the objective function of JPR, $\mathcal{L}_{JPR}(\theta)$, is defined as follows:

$$\mathcal{L}_{JPR}(\theta) = \min_{\theta} \sum_{r_{uij} \in P} y(\hat{r}_{uij}) + \sum_{u \in U, i \in I_u^+} \frac{1}{n_u} (\mathcal{H}_u(i))^2 + \mathcal{R}(\theta) \quad (9)$$

Hence, the JPR loss function sums the CPR loss and the Push loss and a regularization term. To minimize this loss function, we again use stochastic gradient descent (SGD) and the learning algorithm is summarized in the Algorithm 1.

Algorithm 1 Learning Algorithm for JPR

Random initialize θ
repeat
  repeat
    Randomly pick $u \in U$, $i \in I_u^+$ and $j \in I_u^+$
    update θ using equation 4
  until *convergence or max-iteration has been reached;*
  repeat
    Randomly pick $u \in U$ and $i \in I_u^+$
    update θ using equation 8
  until *convergence or max-iteration has been reached;*
until *convergence or max-iteration has been reached;*

In the above algorithm, the model's parameters θ are shared by the CPR and Push models. The algorithm first initializes the parameters and then learns them jointly by alternatively updating them until convergence or the maximum number of iterations is reached.

This learning model can be easily adapted to many RS applications. We can derive from item clicks or views relative preferences [8], i.e., by assuming that what is clicked or viewed is preferred to items not yet considered by the user, while from item that are purchased and rated we can derive more reliable absolute preferences.

We note that once the parameter θ are learned, one can predict the preference scores of user u on a novel item i as $b_i + p_u^T q_i$ and a personalized ranking list of items can be recommended to u by sorting the items according to their predicted scores.

3 EXPERIMENTAL STRATEGY

3.1 Experimental Setup

To measure the performance of the proposed JPR model we need to derive from observational data both pairwise comparisons and ratings of users. Hence, in our experiments we interpret some of the user data as relative and other as absolute preferences, as it is detailed below.

We use three real-world data sets: BookCrossing [10], MovieLens 1m[1] and XING[2]. Using the original data we generate both pairwise comparisons and ratings. The BookCrossing data set contains both implicit (item clicks) and explicit preferences of a user in the form of ratings in a 1-10 scale. We consider the items with rating larger than 7 as relevant and the remaining ones as irrelevant. Next, we derive pairwise comparisons by stating that an item with an implicit preference (click) is preferred to an item that is irrelevant to the user, as it is done in [8].

The Xing dataset records interactions performed by users on job postings. The different types of interaction are: 'click', 'bookmark', 'apply' and 'delete'. We consider the items with 'apply' interaction as relevant and the items with 'delete' as irrelevant. Both click and bookmark are used as implicit signs of preference and from them we derive pairwise comparisons by imposing that the clicked and bookmarked items are preferred to the delete ones.

For MovieLens 1M, we consider the items with 4-5 star ratings as relevant and those rated 1-3 as irrelevant. We again use all the user ratings to derive pairwise comparisons by subtracting two ratings of a same user if they have different rating value. Table 1 summarizes the used data sets and their important features.

Table 1: Dataset characteristics

Data set	BookCrossing	MovieLens-1m	XING
users	271,379	6,040	784,687
items	278,858	3,952	1,029,480
comparisons	5,893,374	104,931,478	180,601,043
ratings	1,048,576	1,000,209	8,826,678
relevant items	228,920	404,925	423,680

We note that XING and Movielens are time stamped and in order to create meaningful training/validation/test splits for these data, we order all the user-item interactions by time-stamp and take the first 80% as training data. Out of this 80%, we randomly select 10% as validation set. For the remaining 20% of the data, we only keep the users and items that appear in the training and validation sets to obtain the test set. For BookCrossing, which is not time-stamped, we randomly split the observed user item interactions into training/validation/test sets with 70/10/20 proportions. The validation sets are used for selecting the best model parameters. We fixed the learning rate to 0.01 and varied the regularization coefficient in the set $\{0.5, 0.05, 0.005, 0.25, 0.025, 0.0025\}$, latent factor size in the set $\{20, 50, 100, 250, 500\}$ and number of iteration in the set $\{50, 100, 500, 1000\}$. We trained all the models till maximum iteration and we repeated our experiment five times; we report average values on the (actual) test set over the runs.

We used four widely-adopted ranking metrics for evaluation: (a) Recall, (b) Mean Average Precision (MAP), (c) Normalized Discounted Cumulative Gain (NDCG), and (d) Mean Reciprocal Rank.

We have compared the proposed ranking model to four baseline algorithms. The first one is Bayesian Personalized Ranking

[1]https://grouplens.org/datasets/movielens/
[2]https://github.com/recsyschallenge/2016

Table 2: Recommendation performance results for various metrics. The best results are in bold (an asterisk means that the model is significantly worse than JPR, p<0.05). We also report the improvement of JPR model over other baseline methods.

Data set	Method	Recall@10	improv.	NDCG@10	improv.	MAP@10	improv.	MRR@10	improv.
MovieLens	BPR-MF	0.0387*	29%	0.7344*	14%	0.0876*	18%	0.0645*	10%
	CPR	0.0419*	19%	0.7708*	8%	0.0933*	11%	0.0672*	6%
	Push	0.0420*	19%	0.7346*	13%	0.0868*	19%	0.0623*	14%
	NN-GK	0.0458*	9%	0.7560*	10%	0.0953*	9%	0.0682	4%
	JPR	**0.0501**	-	**0.8373**	-	**0.1039**	-	**0.0715**	-
BookCrossing	BPR-MF	0.0219	1%	0.0701*	16%	0.0880*	7%	0.0410*	9%
	CPR	0.0218	1%	0.0698*	16%	0.0910	3%	0.0420*	7%
	Push	0.0183*	21%	0.0687*	18%	0.0670*	40%	0.0330*	36%
	NN-GK	0.0172*	29%	0.0775	5%	0.0900	4%	0.0380*	18%
	JPR	**0.0222**	-	**0.0814**	-	**0.0940**	-	**0.0450**	-
XING	BPR-MF	0.0230*	19%	0.0146*	15%	0.0550*	21%	0.0210*	9%
	CPR	0.0260*	5%	0.0163	3%	0.0660	2%	0.0220	4%
	Push	0.0116*	130%	0.0096*	75%	0.0430*	55%	0.0140*	64%
	NN-GK	0.0166*	65%	0.0105*	59%	0.0460*	45%	0.0148*	55%
	JPR	**0.0274**	-	**0.0168**	-	**0.0670**	-	**0.0230**	-

(BPR-MF) which is a popular ranking algorithms that was applied to implicit data [8]. Here, BPR uses only the pairwise comparisons, hence, it should have performances comparable to that of CPR. The second baseline is NN-GK [5], a nearest neighbor based model that uses both ratings and pairwise comparisons, hence it should be comparable with JPR. NN-GK converts all absolute preferences (ratings) into pairwise comparisons and combine them with available pairwise comparisons to predict missing pairwise comparisons and generate a ranking.

3.2 Evaluation Results

The ranking performance of JPR[3] and the baseline approaches is shown in Table 2: for all the metrics, higher values denote better performance. These results show that the proposed model, JPR, has better performance than the baseline models; it has better ranking accuracy for all the considered data sets across all the metrics.

It is also interesting to note that the CPR and Push models when trained separately, offer a ranking performance that is worse than their combination in JPR. Moreover, among the two, CPR has a better performance on all the data sets.

Furthermore, even though both NN-GK and JPR use ratings and pairwise comparisons to compute recommendations, JPR has better performance than NN-GK because it uses pairwise comparisons and ratings separately to learn the target ranking and can more easily optimize their usage. We also note that CPR and BPR-MF are similar ranking models, exploiting relative feedback data only. Both models use a pairwise ranking loss, which is tailored to relative feedback. But, CPR is able to achieve a better ranking accuracy than BPR-MF. This suggests that the usage of cross entropy as rank loss function is preferable.

4 CONCLUSION AND FUTURE WORK

In this paper, we have proposed a ranking model that exploits both relative feedback data (pairwise comparisons) and absolute feedback data (ratings). We have proposed a joint loss function that models both types of preference data. Our experiment results show that the proposed model has a better ranking accuracy compared to state of art algorithms and also show that relative and absolute feedback data should be jointly combined in a proper way in order to optimize the final ranking.

In our future work, we want to better investigate alternative combinations of relative and absolute preferences and to develop mixed active learning strategies [9] that can propose to the users specific items to rate and item pairs to compare, in an optimal way for the target performance metrics.

[3] MovieLens: $d = 100$, $\lambda = 0.05$ and iterations = 100, BookCrossing: $d = 50$, $\lambda = 0.025$ and iterations = 100, XING: $d = 20$, $\lambda = 0.025$ and iterations = 500.

REFERENCES

[1] Iman Barjasteh, Rana Forsati, Abdol-Hossein Esfahanian, and Hayder Radha. 2015. Semi-supervised Collaborative Ranking with Push at Top. (2015).

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

[3] Saikishore Kalloori and Francesco Ricci. 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.

[4] Saikishore Kalloori, Francesco Ricci, and Rosella Gennari. 2018. Eliciting pairwise preferences in recommender systems. In *Proceedings of the 12th ACM Conference on Recommender Systems*. ACM, 329–337.

[5] Saikishore Kalloori, Francesco Ricci, and Marko Tkalcic. 2016. Pairwise preferences based matrix factorization and nearest neighbor recommendation techniques. In *Proceedings of the 10th ACM Conference on Recommender Systems*. ACM, 143–146.

[6] Hongzhi Liu, Zhonghai Wu, and Xing Zhang. 2018. CPLR: Collaborative pairwise learning to rank for personalized recommendation. *Knowledge-Based Systems* 148 (2018), 31–40.

[7] Jingzhou Liu, Wei-Cheng Chang, Yuexin Wu, and Yiming Yang. 2017. Deep learning for extreme multi-label text classification. In *Proceedings of the 40th International ACM SIGIR Conference*. ACM, 115–124.

[8] Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. 2009. BPR: Bayesian personalized ranking from implicit feedback. In *Proceedings of the twenty-fifth conference on uncertainty in artificial intelligence*. AUAI Press.

[9] Anna Sepliarskaia, Julia Kiseleva, Filip Radlinski, and Maarten de Rijke. 2018. Preference elicitation as an optimization problem. In *Proceedings of the 12th ACM Conference on Recommender Systems*. ACM, 172–180.

[10] Cai-Nicolas Ziegler, Sean M McNee, Joseph A Konstan, and Georg Lausen. 2005. Improving recommendation lists through topic diversification. In *Proceedings of the 14th international conference on World Wide Web*. ACM.