# Pairwise Preferences Elicitation and Exploitation for Conversational Collaborative Filtering

Laura Blėdaitė*
Faculty of Computer Science
Free University of Bozen - Bolzano
Piazza Domenicani 3, I - 39100, Bolzano
laura.bledaite@gmail.com

Francesco Ricci
Faculty of Computer Science
Free University of Bozen - Bolzano
Piazza Domenicani 3, I - 39100, Bolzano
fricci@unibz.it

## ABSTRACT

The research and development of recommender systems is dominated by models of user's preferences learned from ratings for items. However, ratings have several disadvantages, which we discuss, and in order to address these issues we analyse another way to articulate preferences, i.e., as pairwise comparisons: item A is preferred to item B. We have developed a recommendation technology that, combining ratings and pairwise preferences, can generate better recommendations than a state of the art solution uniquely based on ratings.

## Categories and Subject Descriptors

H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval—*Information Filtering*

## Keywords

Pairwise preferences; collaborative filtering; recommender systems

## 1. INTRODUCTION

Recommender systems (RSs) are popular Web applications that generate personalised recommendations for items that are estimated to be relevant and useful for their target users [20]. The research and application of RSs is dominated by the usage of ratings, which indicate absolute preferences for items. In its core computational step, e.g., by using collaborative filtering [3], a RS builds a prediction model that, analysing the available ratings, estimates unknown ones.

However, ratings have several disadvantages. First of all, they must be expressed in a predefined scale, which has its own characteristics, and measures taken according to a scale cannot be easily converted to another one [6]. Hence, choosing the right scale is always an issue. Moreover, since ratings represent evaluations measured against an absolute benchmark, it could be difficult for the user to consistently rate items. For instance, if a user rates an item with the highest value and succesively finds another item which she likes more, then there is no way to express such a preference.

Considering these issues, we have analysed another way to express preferences, i.e., as pairwise comparisons of alternative options, such as, A is preferred to B. However, whether entering pairwise preferences is easier than ratings is debatable. In [9] the authors claim that it is easier to decide which item is preferred among two, rather than rating them in some predefined scale. Conversely, in [17] comparing alternative interfaces for rating and ranking the authors conclude that "rating is the more familiar and less cognitively demanding form of judgement", and found that a rating interface, with the additional support of showing one example item for each star level, was preferred to an interface supporting pairwise comparisons of items. Moreover, while pairwise preferences have been studied in the learning to rank literature [5, 19, 8], they have been rarely used for building RSs (in combination with ratings).

Working on the proposition that pairwise preferences might provide a viable complement to ratings in RSs, we have developed a recommendation technology that combines ratings and pairwise preferences to model user preferences and to generate recommendations. We have compared that solution with a state of the art rating-based approach (based on matrix factorization [11]), and validated the following hypotheses:

1. Pairwise preferences can be as easy to enter as ratings (provided that an effective interface is built);
2. Pairwise preferences can help users more than ratings to understand their preferences;
3. The proposed pairwise-based recommendation technology has a better accuracy and ranking quality;

The rest of this article is organised in the following way. In Section 2 we illustrate the implemented preference acquisition interaction and we describe the implemented ranking and recommendation technique. In Section 3, the evaluation strategy is described and in Section 4 the results are presented. Finally, we discuss some related work and draw the conclusions of our research.

## 2. PAIRWISE-BASED RECOMMENDER

In order to validate our research hypotheses, we have implemented two movie recommender systems: RAO (RAtings Only) which is based on ratings and Matrix Factorization (MF) (SVD method [11]), and PPR (Pairwise Preferences and Ratings) which analyses user preferences in the form of pairwise preferences (pair-scores) and makes recommendations using them together with a possibly pre-existent ratings data set. In this section we illustrate the important features of these systems.

*Current affiliation: Twitter Inc.


*HT '15,* September 1–4, 2015, Guzelyurt, Northern Cyprus.

DOI: http://dx.doi.org/10.1145/2645710.2645757.

## 2.1 Preference Acquisition

Preference elicitation requires a suitable GUI. We have implemented a standard five-stars rating interface for the RAO system. Figure 1 instead shows the GUIs that we designed for the PPR system. It enables the users to compare pairs of items and to enter to what extent an item is preferred to another (pair-score). We decided to use a slider: the closer the slider pointer is dragged to an item the more this item is preferred to the other.

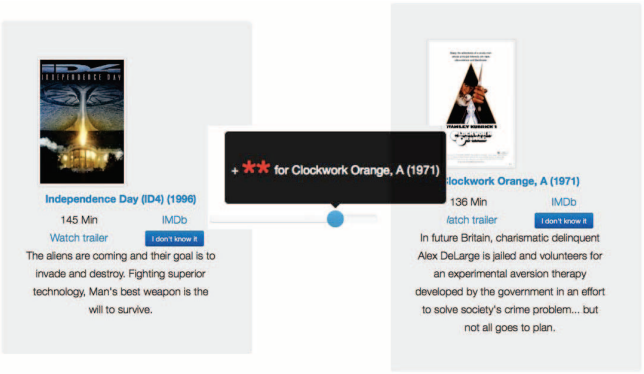


**Figure 1: Items' comparison interface producing pair-scores**

The main problem that arises in system controlled preference acquisition is the choice of the items, movies in our recommender system, to be shown to the user to rate (rating scenario), or the choice of the movie pairs to compare (pairwise preferences scenario). This is an active learning problem that has received already some attention in the RS community [21, 4]. "Active learning" means that the system actively decides what (preference) data to acquire before starting the learning phase, which in RSs is the rating prediction model building.

For ratings' acquisition in RAO, we adopted a variation of a popular active learning strategy, namely: $log(pop)entropy$ [18]. But, we replaced entropy with variance, because variance is a diversity measure for ordinal data, hence, it is more suitable for measuring the rating diversity. The chosen scoring measure for ranking the items to present to the user to rate - $LpopVar$ - is given in Equation 1. It scores and ranks higher the most popular movies ($log$ factor) with the most diverse ratings ($variance$ factor). The higher the score for item $i$, the higher $i$ is ranked in the list of items presented to the user to rate. We denote with $u \in U$ a user, with $i \in I$ an item, and with $r_{ui}$ the rating that $u$ gave to $i$. $U_i$ is the set of users that rated item $i$, and $\bar{r}_i$ is the average of the ratings of item $i$.

$$LpopVar(i) = log(|U_i|) \left( \frac{1}{|U_i|} \sum_{u \in U_i} (r_{ui} - \bar{r}_i)^2 \right) \quad (1)$$

While the literature offers several options for the selection of the item to ask the user to rate, acquiring pairwise preferences in recommender systems has not been considered so far. This is challenging since the number of potential pairs of items that the user could compare is quadratic in the number of items and without some well designed system support, in the form of a selection or ranking of item pairs, the task would be hard to complete. Hence, we have introduced a scoring function, which is analogous to $LpopVar$ and it is shown in Equation 2. It is used in the PPR system for ranking the item pairs to be presented to the user to compare.

$$log(|U_i|)\ log(|U_j|)\ (1 - \rho_{ij}) \quad (2)$$

Here, $\rho_{ij}$ denotes the Pearson correlation between the ratings of the items $i$ and $j$ expressed by the users in $U_{ij} = U_i \cap U_j$, i.e., the users that rated both $i$ and $j$. Items' popularity is considered as in the rating scenario, but we also incorporate a measure of the decorrelation of the ratings of the items in the pair $(1 - correlation)$. The formula implements the heuristics that the more de-correlated the ratings of the two items are, the more the user preference for one of the two will help the system to understand the user tastes. Using this ranking approach the same item may be shown several times in different pairwise comparisons.

## 2.2 Pairwise-based Recommendations

With a collection of ratings or/and pairwise preferences the PPR system uses a recommendations' ranking technique described in this section. We first illustrate a "non personalised" ranking method based on the pairwise comparison of items in the form of pair-scores [12]. Then, we describe our original modification aimed at obtaining a personalised ranking.

If a set of ratings is available, a skew-symmetric $n \times n$ matrix $K$ is defined, where $n = |I|$, and with entry $k_{ij}$ as in Equation 3.

$$k_{ij} = \frac{1}{m_{ij}} \sum_{u \in U_{ij}} r_{uij} \quad (3)$$

$$r_{uij} = r_{ui} - r_{uj} \quad (4)$$

where $U_{ij} = U_i \cap U_j$ is the set of users that rated both $i$ and $j$, and $m_{ij} = |U_{ij}|$. In [12] (chapter 9) the following "scoring" vector $\nu = (\nu_1, \ldots, \nu_n)$ is computed:

$$\nu_i = \frac{\sum_{j=1}^{n} k_{ij}}{n} \quad (5)$$

The entries of $\nu$ are obtained by simply averaging the rows of the matrix $K$. It is shown that these entries determine a ranking of the items with the property that $\nu_i - \nu_j$ gives the best approximation of $k_{ij}$, i.e., the difference of the $\nu$ scores of two items tells us how much on average an item receives more star ratings than the other. It is worth noting that such a ranking can be obtained even if in the $K$ matrix there are conflicting preferences such as: $k_{ij} > 0$, $k_{jl} > 0$ and $k_{li} > 0$, i.e., item $i$ is preferred to $j$, $j$ is preferred to $l$, but also $l$ is preferred to $i$.

Our personalised version of this ranking technique, which is illustrated below, incorporates user-to-user similarity weights in the computation of $K$, hence computing a $K(u)$ matrix for each user $u$ and then producing a personalised ranking of the items using again the formula 5. Namely, $k_{uij}$, the entries of $K(u)$, are the system predictions of how much the user $u$ will prefer $i$ over $j$. Hence, while in ratings based systems one predicts ratings, in pairwise preferences approaches [5] one first estimates how much a user likes an item more that another and then aggregates these predictions in the final ranking function, as in Equation 5.

We note that the user ratings for two items can be easily converted into a pair score, as it is shown in equation 4. But also, with the help of the slider-based GUI shown in Figure 1, we are able to directly collect pair-scores. When the user $u$ moves the slider towards item $i$, this means that $u$ prefers $i$ to $j$ ($i \succ_u j$) and he can also select how much $i$ is preferred to $j$, hence we can assign a positive value to $r_{uij}$. While, if user $u$ moves the slider towards $j$ a negative value is assigned to $r_{uij}$ ($i \prec_u j$) (see Equation 6).

We decided to collect pair scores in the range $[-4, +4]$ to be able to exploit a collection of pre-existent ratings in the $[1, 5]$ scale. But the method described here can be used without any modification,

even when the pair scores are in the range $\{-1, 0, 1\}$, that is, when the user simply states that she prefers an item to another or says that the two are equally preferred.

$$r_{uij} = \begin{cases} \in \{4,3,2,1\} & i \succ_u j \\ 0 & \text{no preference} \\ \in \{-1,-2,-3,-4\} & i \prec_u j \end{cases} \quad (6)$$

Hence, in order to generate personalised recommendations using a collection of ratings and pair-scores the system converts all the available ratings (if there are any) in pair-scores, adds the available pair-scores, and then for the target user $u$, the personalised values of the $K(u)$ matrix are calculated as follows:

$$k_{uij} = \frac{1}{\sum_{v \in U_{ij}} w'_{u,v}} \sum_{v \in U_{ij}} w'_{uv}\, r_{vij} \quad (7)$$

where the user-to-user similarity $w'_{uv}$, as defined in Equation 8, is a generalisation to pair scored of the original Pearson correlation defined on ratings [3]. Actually, it is the Pearson correlation computed among the users' pair-scores, multiplied by a significance score:

$$w'_{uv} = \frac{min\left(|\mathcal{I}_{uv}|, \gamma\right)}{\gamma}\, w_{uv} \quad (8)$$

$$w_{uv} = \frac{\sum_{(i,j) \in \mathcal{I}_{uv}} (r_{uij} - \bar{r}_u)(r_{vij} - \bar{r}_v)}{\sqrt{\sum_{(i,j) \in \mathcal{I}_{uv}} (r_{uij} - \bar{r}_u)^2 \sum_{(i,j) \in \mathcal{I}_{uv}} (r_{vij} - \bar{r}_v)^2}} \quad (9)$$

Here $\bar{r}_v$ is the user's $u$ average of all pairwise preferences, and $\mathcal{I}_{uv}$ is the set of all pairs $(i, j)$ of items that both user $u$ and user $v$ rated (or compared), and such that $i < j$. The significance score $\frac{min(|\mathcal{I}_{uv}|,\gamma)}{\gamma}$ decreases the similarity $w_{uv}$ when $|\mathcal{I}_{uv}|$ is smaller that $\gamma$, i.e., when users $u$ and $v$ compared few common pairs of items. $\gamma$ is a parameter that must be cross-validated. In our experiments we obtained the best performance for $\gamma = 7$.

# 3. EXPERIMENTAL STRATEGY

We recall that we have implemented two fully operational recommender systems that interact with the users, acquire preferences (ratings or pairwise comparisons), and rank items in order to select the top-n recommendations for the users: RAO - which is based on RAtings Only and uses Matrix Factorization; PPR - which is based on a mixture of Pairwise Preferences acquired during the interaction with the users and possibly pre-existent Ratings. By using the two mentioned systems we have validated our research hypotheses by performing a live user study, as an A/B test (between group).

The initial data set of ratings is common for both systems and contains those for the top 100 movies scored by the $LpopVar$ criterion (Equation 1) that are present in the MovieLens 100K data set (http://grouplens.org/datasets/movielens/).

The evaluation strategy of the systems, included the following stages and steps:

- Stage 1: Initial preference elicitation
  - User preference elicitation (ratings or pairwise preferences);
  - User evaluation of the preference elicitation procedure (questionnaire).
- Stage 2: Recommendation and preference revision
  - Recommendation presentation and user assessment of a first set of recommendations I;
  - User input of additional preferences;
  - Recommendation presentation and user assessment of a second set of recommendations II;
  - User evaluation of the recommendations quality (questionnaire).

The above listed steps are further described in the following. The users were recruited for the experiment mostly by using social media channels and e-mail address lists. Many of them are aged between 25 to 35. A high percentage of them are either undergraduate, graduate, PhD students, recent graduates or university staff. We think that the sample is quite representative of the real users of such a movie recommender system, and more in general for such type of systems.

There were 97 users registered to the experiment. However, not all the users finished the whole experiment (precise numbers are given later). During the initial "preference elicitation" stage of the experiment, user preferences were gathered in the form of either ratings or pairwise preferences, depending on the system to which the users were assigned. Items to rate or item pairs to compare where ranked and presented using the active learning technology that is described in Section 2.1. In case a user was assigned to the RAO system, she was asked to provide ratings for the items. In case she was assigned to the PPR system, she was asked to provide pairwise preferences (Figure 1). We did not ask users to provide a precise number of ratings or pairwise preferences, we simply let the user add as many preferences as she liked. In fact, we were interested in measuring the effort that users freely decide to devote to preference elicitation, estimated as the number of inserted preferences. Preference elicitation is typically seen by the users as a burden, hence we wanted to understand which preference elicitation method may be better accepted and used by the users.

In the second step of the first stage, users were asked to evaluate the preference elicitation process. 89 users completed the first stage (RAO 44 and PPR 45) and answered to the following questionnaire on the preference elicitation process:

1. I have fun using the system;
2. Using the system is a pleasant experience;
3. The system makes me more aware of my choice options;
4. I feel bored when I am using the system.

We took these questions from a survey designed for measuring the perceived system effectiveness and fun that was elaborated by Knijnenburg et al. [10].

The second stage of the experiment (recommendation and preference revision) was run after 15 days, when all the 89 users that accessed the system in the first stage did complete the preference elicitation process. The 15 days interruption between the two stages is not deemed as problematic. It is a common practice to enter ratings in a session (stage 1 of the experiment) and to request recommendations subsequently (the user task in the second stage of the experiment).

In the first step of the second stage of the experiment, the users were given top 5 recommendations displayed in a list (Figure 2). Ranking of the recommendations were computed using SVD matrix factorization for RAO [11] and using the proposed technique for PPR, and were based on the preferences that all the users provided during the first stage (plus the ratings for the selected 100

items that were already present in MovieLens). While browsing the recommendations the users could watch a trailer of the recommended movie or access the corresponding IMDb page. Moreover, users were asked to mark the items that they considered "good recommendations" and the ones they "have seen". This information enabled us to compare the accuracy of the two recommendation processes in terms of precision, which is calculated as the proportion of the relevant items (good recommendations) among the 5 recommended items. Moreover, in order to assess the quality of

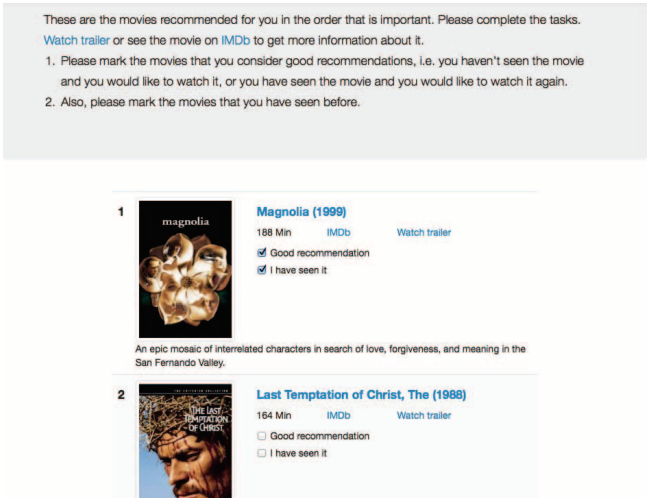


**Figure 2: List of Recommendations**

the systems' generated rankings we used normalized Discounted Cumulative Gain ($nDCG$) [15, 3], which is a popular and well accepted measure of ranking quality.

In the second step of the second stage the users were asked to provide additional preferences (ratings or pairwise preferences depending on the system they were assigned to) about the items that they marked as seen in the previous step (recommendation I), by using again the same preference elicitation interface that they used in the first step of the first stage. Note that our ranking approach can handle preferences that are possibly conflicting with those already expressed (see Section 2.2). The goal of this step was to involve the user by interacting with the system in order to offer her better and better recommendations.

Next, the users were given an improved set of recommendations based on all the entered preferences (both in the first and second stage) (again as in Figure 2). They marked again the good and the seen recommendations. This approach enabled us to test the improvement of the accuracy of the two recommendation processes after additional preferences, using both approaches, were acquired. We measured again precision and $nDCG$ for both system.

We conjectured that the subsequent recommendation list, which is computed by using also the additional preferences collected on items belonging to the first recommendation list, could be more accurate than the first list, and the usage of pairwise preference could improve more the recommendation accuracy (precision and nDCG).

Finally, the users were asked to provide an overall feedback to the recommendations quality in general. The feedback was collected using a questionnaire (answers in a Likert scale). This questionnaire was already used in similar experiments [10]:

1. I liked the items recommended by the system;
2. The recommended items fitted my preference;
3. The recommended items were well-chosen;
4. The recommended items were relevant;
5. The system recommended too many bad items;
6. I didn't like any of the recommended items;
7. The items I selected were "the best among the worst".

69 (34 from RAO and 35 from PPR) users finished both stages of the experiment. Thus, 69 users filled in the questionnaire at the end of the second stage of the experiment. We use this data to compare their perceived recommendation quality. We note that among these 69 users only 30 provided additional ratings or preferences (16 using system RAO and 14 using system PPR), i.e., 39 skipped this phase. Thus we have used only this data when comparing recommendation prediction accuracy and ranking quality before/after additional preferences were entered.

# 4. RESULTS

Analysing the replies of the users to the first questionnaire on the satisfaction for the preference elicitation process we found a larger overall score of $66.66$ for PPR as compared to $62.78$ for RAO (this validates the fist research hypothesis listed in the Introduction). This score is computed by taking each reply to the four given statements and converting them into a score in the range 0-4. For sentences where larger agreement means a higher perceived satisfaction, i.e., the first three, the score contribution is the scale position minus 1. For sentences where larger agreement means a lower perceived satisfaction (the fourth) the contribution is 5 minus the scale position. The resulting score was then scaled to [0, 100].

We also discovered that the system that elicits pairwise preferences makes users more aware of their choice options, i.e., users replied to question 3 with a much larger agreement. We conducted Mann-Whitney and Kruskal-Wallis tests and both of them proved significant difference in favour of PPR (both p-values $\approx 0.0001$). This proves the second research hypothesis that we made.

When analysing the scores for separate questions of the questionnaire about the perceived recommendation quality, which the user gave at the end of the complete recommendation process, i.e., end of stage 2, we observed a clear tendency of PPR to outperform RAO. The pooled score, which was computed similarly to the score of the first questionnaire on the user satisfaction of the preference elicitation process, for RAO is $61.34$ and for PPR is $63.27$. However, because of a small sample size, a significant difference was not observed.

Discussing now the recommendation accuracy of the two systems, Table 1 shows $nDCG$ and precision of RAO and PPR before and after additional preferences were entered in stage two and p-values of the tests for significance of $nDCG$ and precision differences: a) between systems before and after the additional preferences; and b) within each system before versus after additional preferences. We note that we could compute these evaluation metrics because the users were asked to mark the recommendations that they considered as good. Hence, we stress that the results shown here are not offline estimations of precision and nDCG, but the effective performance of the system recommendations as evaluated by the users.

As it can be seen from Table 1, before the additional preferences are entered, i.e., when comparing the initial recommendation lists, PPR performs significantly better than the RAO in terms of $nDCG$ (p-value $= 0.024$). PPR has also a better precision than RAO, but in this case the difference is not significant. After the additional

**Table 1: Recommendation accuracy and its improvement after additional preferences were entered**

| | nDCG | | | precision | | |
|---|---|---|---|---|---|---|
| | before | after | p-value | before | after | p-value |
| RAO | 0.54 | 0.66 | 0.180 | 0.45 | 0.49 | 0.346 |
| PPR | 0.79 | 0.84 | 0.295 | 0.51 | 0.64 | 0.035 * |
| p-value | 0.024 * | 0.046 * | | 0.35 | 0.044 * | |

preferences are entered by the users, i.e., when comparing the improved recommendation lists presented by the two systems, PPR performs significantly better than RAO in terms of both $nDCG$ (p-value $= 0.046$) and precision (p-value $= 0.044$).

Table 1 also shows, as expected, that there is always an improvement in precision and $nDCG$ for both systems after the user has provided additional preferences. But, there is a significant improvement of recommendation accuracy, in terms of precision, only after additional preferences were entered in PPR (p-value $= 0.035$). These results prove the third hypothesis that we made, i.e., the usage of pairwise preferences, compared to the exploitation of ratings, improves more the recommendation accuracy and ranking quality.

We also looked at the number of preferences entered in the systems in the two stages. In the first stage of the experiment, $1,415$ ratings and $2,262$ pairwise preferences were collected. In the second stage, on average per user, $2.06$ additional ratings and $1.93$ additional pairwise preferences were provided by the users using RAO and PPR, respectively. Hence, there was a difference in terms of the number of preferences entered by the users using the two systems; it is noteworthy that more pairwise preferences than ratings were acquired in the first stage. We can conclude that overall these results confirm our main research hypothesis, that is, pairwise preferences are a viable approach to preference elicitation and the generated recommendations are even superior to those produced by Matrix Factorization.

# 5. RELATED WORK

In [2] the authors discuss issues related to rating inconsistency and the user difficulty in mapping preferences to ratings, while [16] addresses these problems by introducing improved user interfaces to support the preference to rating mapping process. The suggested methods include personalised tags and exemplars to relate rating decisions to prior ones. It has been concluded that, notwithstanding the usefulness of their proposed solutions, it remains hard for the user to enter ratings.

The authors of [9] have already guessed that pairwise preferences are easier to formulate and to reason about than ratings. Namely, it is easier to decide which item (and how much more) is preferred to another, rather than to rate both items in some arbitrary scale, e.g., the common 5-star scale.

Besides, in [17] the authors compare alternative interfaces for rating and ranking by measuring the user perceived: speed, accuracy, mental demand, suitability for organization, fun to use, and overall preference. In that study the user task was to rank 20 movies. It is worth noting that in their scenario they derive results that are very different from ours. For instance, they found that a rating interface, with the additional support of showing one example item for each star level, was preferred to an interface supporting pairwise comparisons of items. This diversity stresses the importance to evaluate preference elicitation interfaces in the context of their usage, since one cannot derive absolute measures of a goodness of an approach without embedding it in a fully operational system.

In the RS literature some formal ranking models based on pairwise preferences exist [23, 7, 22, 19]. However, none of them has been developed, together with an appropriate GUI, for supporting the full interaction of the user with the system: preference elicitation, preference revision, and recommendation browsing.

A notable example of a recommender system based on pairwise preferences is described in [13]. However, there are several differences between this approach and that one presented in this paper. For instance, in our system the user is entering preferences by comparing pairs of movies while in [13] the user is asked to compare sets of movies. Moreover, the number of comparisons in our case is essentially not limited by any condition and we have developed a novel active learning strategy for helping the user to compare items. In [13] the number of comparisons is equal to the number of factors of the used matrix factorization model, which is necessary in order to bootstrap the approach. For that reason, the factor model must be rather simple, since for each factor the system generates a preference elicitation comparison of two sets of movies. It is unlikely that a user can go through many of these questions, while an accurate factor model can require a very large number of factors (hundreds, thousands and even more).

# 6. CONCLUSION AND FUTURE WORK

By conducting an online A/B test, we have shown that it is possible to build recommender systems that incorporate pairwise preferences and perform better than state of the art rating-only based solutions (in terms of recommendation accuracy measured by $nDCG$ and precision). We have also shown that such type of systems can improve more the recommendation accuracy after the user provides additional preferences. Additionally, we have shown that asking a user to compare movies makes her more aware of her choice options, and by doing that the system is able to collect more preferences (pairwise comparisons vs. ratings). We have therefore validated the research hypotheses that we stated in the Introduction of this paper.

We want here to mention some limitations of the presented work and suggest some branches of further research. First of all, a deeper analysis of the pairwise preference request generation is required. In that respect, further work is needed in the development of effective user interfaces that make use of active learning strategies for the PPR model and may elicit mixed preference data, i.e., both ratings and pairwise preferences. This is especially important when there are no pre-existent ratings (cold-start), as we have assumed in our experiments, and it is therefore important to acquire users' preference information efficiently. Another important research line is the better usage of session data in PPR. In fact, preference elicitation is strongly influenced by the interaction context which varies at each single session [14, 1].

Finally we must explicitly note that the proposed ranking method illustrated in Section 2.2 is just one possible solution for the considered ranking problem. We imagine that other label ranking techniques could be applied to recommender systems and we believe that in the future more research works could be dedicated to this interesting topic.

# 7. REFERENCES


[1] G. Adomavicius, B. Mobasher, F. Ricci, and A. Tuzhilin. Context-aware recommender systems. *AI Magazine*, 32(3):67–80, 2011.

[2] A. Bellogín, A. Said, and A. P. de Vries. The magic barrier of recommender systems - no magic, just ratings. In *User Modeling, Adaptation, and Personalization - 22nd International Conference, UMAP 2014, Aalborg, Denmark, July 7-11, 2014. Proceedings*, pages 25–36, 2014.

[3] C. Desrosiers and G. Karypis. A comprehensive survey of neighborhood-based recommendation methods. In F. Ricci, L. Rokach, B. Shapira, and P. B. Kantor, editors, *Recommender Systems Handbook*, pages 107–144. Springer, 2011.

[4] M. Elahi, F. Ricci, and N. Rubens. Active learning in collaborative filtering recommender systems. In *E-Commerce and Web Technologies - 15th International Conference, EC-Web 2014, Munich, Germany, September 1-4, 2014. Proceedings*, pages 113–124, 2014.

[5] J. Fürnkranz and E. Hüllermeier. Preference learning and ranking by pairwise comparison. In J. Fürnkranz and E. Hüllermeier, editors, *Preference Learning*, pages 65–82. Springer Berlin Heidelberg, 2011.

[6] C. Gena, R. Brogi, F. Cena, and F. Vernero. The impact of rating scales on user's rating behavior. In *User Modeling, Adaption and Personalization - 19th International Conference, UMAP 2011, Girona, Spain, July 11-15, 2011. Proceedings*, pages 123–134, 2011.

[7] D. F. Gleich and L.-H. Lim. Rank aggregation via nuclear norm minimization. In *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Diego, CA, USA, August 21-24, 2011*, pages 60–68, 2011.

[8] E. Hüllermeier, J. Fürnkranz, W. Cheng, and K. Brinker. Label ranking by learning pairwise preferences. *Artificial Intelligence*, 172(16-17):1897–1916, Nov. 2008.

[9] N. Jones, A. Brun, A. Boyer, and A. Hamad. An exploratory work in using comparisons instead of ratings. In *E-Commerce and Web Technologies - 12th International Conference, EC-Web 2011, Toulouse, France, August 30 - September 1, 2011. Proceedings*, pages 184–195, 2011.

[10] B. P. Knijnenburg, M. C. Willemsen, Z. Gantner, H. Soncu, and C. Newell. Explaining the user experience of recommender systems. *User Modeling and User-Adapted Interaction*, 22:441–504, 2012.

[11] Y. Koren and R. Bell. Advances in collaborative filtering. In F. Ricci, L. Rokach, B. Shapira, and P. B. Kantor, editors, *Recommender Systems Handbook*, pages 145–186. Springer Science and Business Media, 2011.

[12] A. N. Langville and C. D. Meyer. *Who's #1?: The Science of Rating and Ranking*. Princeton University Press, 2012.

[13] B. Loepp, T. Hussein, and J. Ziegler. Choice-based preference elicitation for collaborative filtering recommender systems. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '14, pages 3085–3094, New York, NY, USA, 2014. ACM.

[14] T. Mahmood and F. Ricci. Improving recommender systems with adaptive conversational strategies. In *HYPERTEXT 2009, Proceedings of the 20th ACM Conference on Hypertext and Hypermedia, Torino, Italy, June 29 - July 1, 2009*, pages 73–82, 2009.

[15] C. D. Manning, P. Raghavan, and H. Schütze. *Introduction to Information Retrieval*. Cambridge University Press, 2008.

[16] T. T. Nguyen, D. Kluver, T.-Y. Wang, P.-M. Hui, M. D. Ekstrand, M. C. Willemsen, and J. Riedl. Rating support interfaces to improve user experience and recommender accuracy. In *Proc. RecSys 2013*, pages 149–156. ACM, 2013.

[17] S. Nobarany, L. Oram, V. K. Rajendran, C.-H. Chen, J. McGrenere, and T. Munzner. The design space of opinion measurement interfaces: Exploring recall support for rating and ranking. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '12, pages 2035–2044, New York, NY, USA, 2012. ACM.

[18] A. Rashid, I. Alberta, D. Cosley, S. Lam, S. McNee, J. Konstan, and J. Riedl. Getting to know you: Learning new user preferences in recommender systems. In *in Proc. of the International Conference on Intelligent User Interfaces*, pages 127–134, 2002.

[19] S. Rendle, C. Freudenthaler, Z. Gantner, and L. Schmidt-Thieme. Bpr: Bayesian personalized ranking from implicit feedback. In *Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence*, UAI '09, pages 452–461, Arlington, Virginia, United States, 2009. AUAI Press.

[20] F. Ricci. Recommender systems: Models and techniques. In R. Alhajj and J. G. Rokne, editors, *Encyclopedia of Social Network Analysis and Mining*, pages 1511–1522. Springer, 2014.

[21] N. Rubens, D. Kaplan, and M. Sugiyama. Active learning in recommender systems. In F. Ricci, L. Rokach, B. Shapira, and P. Kantor, editors, *Recommender Systems Handbook*, pages 735–767. Springer Verlag, 2011.

[22] Y. Shi, A. Karatzoglou, L. Baltrunas, M. Larson, N. Oliver, and A. Hanjalic. Climf: Learning to maximize reciprocal rank with collaborative less-is-more filtering. In *Proceedings of the Sixth ACM Conference on Recommender Systems*, RecSys '12, pages 139–146, New York, NY, USA, 2012. ACM.

[23] S. Wang, J. Sun, B. J. Gao, and J. Ma. Adapting vector space model to ranking-based collaborative filtering. In *21st ACM International Conference on Information and Knowledge Management (CIKM '12)*, 2012.