Eliciting Pairwise Preferences in Recommender Systems

Saikishore Kalloori, Francesco Ricci and Rosella Gennari
Free University of Bozen - Bolzano, Italy
Piazza Domenicani 3, I - 39100, Bolzano, Italy
ksaikishore@unibz.it,fricci@unibz.it,gennari@inf.unibz.it

ABSTRACT

Preference data in the form of ratings or likes for items are widely used in many Recommender Systems. However, previous research has shown that even item comparisons, which generate pairwise preference data, can be used to model user preferences. Moreover, pairwise preferences can be effectively combined with ratings to compute recommendations. In such hybrid approaches, the Recommender System requires to elicit both types of preference data from the user. In this work, we aim at identifying how and when to elicit pairwise preferences, i.e., when this form of user preference data is more meaningful for the user to express and more beneficial for the system. We conducted an online A/B test and compared a rating-only based system variant with another variant that allows the user to enter both types of preferences. Our results demonstrate that pairwise preferences are valuable and useful, especially when the user is focusing on a specific type of items. By incorporating pairwise preferences, the system can generate better recommendations than a state of the art rating-only based solution. Additionally, our results indicate that there seems to be a dependency between the user's personality, the perceived system usability and the satisfaction for the preference elicitation procedure, which varies if only ratings or a combination of ratings and pairwise preferences are elicited.

CCS CONCEPTS

•Information systems → Recommender systems;

KEYWORDS

Pairwise Preferences; Ratings; Recommender Systems

ACM Reference format:
Saikishore Kalloori, Francesco Ricci and Rosella Gennari. 2018. Eliciting Pairwise Preferences in Recommender Systems. In *Proceedings of Twelfth ACM Conference on Recommender Systems, Vancouver, BC, Canada, October 2–7, 2018 (RecSys '18),* 9 pages.
DOI: 10.1145/3240323.3240364

1 INTRODUCTION

Most of the current Recommender Systems (RSs) generate personalised suggestions by leveraging preference information derived from absolute item evaluations, e.g., user ratings or likes [31]. However, this type of preferences have few disadvantages, illustrated in previous literature [5, 22, 23]. For instance, if a user has assigned the highest rating to an item and, subsequently, the user finds that he prefers another item to the first one, the user has no choice but to also give to this new item the highest rating [11]. Moreover, if most of the items are rated with the largest rating, then it is difficult to understand which items the user does like the most.

For such reasons, few researchers [5, 11, 22] have developed recommendation techniques that leverage an alternative way to collect user preferences, namely, pairwise comparisons, such as item i is preferred to item j. In these scenarios, users provide their preferences by expressing which item is preferred in a pair (i,j). In [24, 25], it was shown that pairwise preferences can be effectively used for training matrix factorization and nearest neighbor approaches, and ultimately to compute effective recommendations.

It is worth noting that pairwise preferences naturally arise and are expressed by users (directly or indirectly) in many decision-making scenarios. Actually, in everyday life, there are situations where rating alternative options is not the most natural mechanism for expressing preferences and making decisions. For instance, we do not rate sweaters when we want to buy one. It is more likely that we will compare them and purchase the preferred one. However, understanding which pairwise preferences the user should express, so as to learn her user model and when to acquire these comparisons, is a pivotal issue in designing effective RSs based on this form of preference data.

For this reason, in this work, we aim at understanding when eliciting pairwise preferences is meaningful and beneficial. We hypothesize that pairwise preferences are more effective, if compared with ratings, in situations and scenarios where the user has a rather clear objective and is looking for a specific type of items (recommendation). For instance, when a user wants to enjoy a beach resort along with his family during a weekend, it is more likely that he will choose the best option by comparing the available ones instead of rating them. We focus on such user situations having a specific goal/context, and we model their preferences. Our intuition is that in order to make a comparison, the user requires a (set of) criteria to formulate a judgment. Moreover, we also hypothesize that the items to be compared should not be too different; a buyer can compare two cars but not a car and a bicycle. In fact, even the adjective "comparable" means that the compared objects are somewhat similar.

We also aim at addressing the issue of which specific items the user should compare. In fact, in addition to the issue mentioned above (i.e., that the compared items should be somewhat similar), an additional downside of acquiring user preference information from pairwise comparison is that the number of possible comparisons in a set of N items is N^2. Even though it has been shown

RecSys '18, Vancouver, BC, Canada

978-1-4503-5901-6/18/10…$15.00
DOI: 10.1145/3240323.3240364

that in practice RSs based on pairwise preferences do not require more preference data than systems based on ratings [5], it is still important and critical to identify, for each user, a convenient set of comparisons that she should express. In order to effectively elicit pairwise preferences, the RS needs a usable GUI and an active learning method that can identify which pairs of items the users could and should compare.

Hence, in this paper, we develop a specific interaction model, GUI components, an active learning strategy for pairwise preferences and a personalized ranking algorithm that uses pairwise preferences in order to validate the following hypotheses:

(H_1) Pairwise preferences can lead to a larger system usability compared to ratings.

(H_2) The usage of pairwise preferences can lead to a larger user satisfaction for the preference elicitation process compared to the usage of ratings, when the choice set has been already reduced.

(H_3) Pairwise preferences can lead to higher RS accuracy and quality compared to ratings when the user has a clear decision making objective.

Modeling user preferences in the form of pairwise preferences and identifying which situation is best suited for eliciting pairwise preferences has not been explored in RSs so far. Therefore, in order to validate our research hypotheses, we have implemented techniques for pairwise preference elicitation and recommendation generation in a mobile RS, called South Tyrol Suggests (STS). STS recommends interesting Places of Interests (POIs) in South Tyrol region in Italy [7]. STS is an Android-based RS that provides users with context-aware recommendations. We have implemented two RSs variants, using STS, and conducted a live user study. One variant is based on ratings only while the other employs our newly developed algorithms, gathers user preferences in the form of pairwise preferences and makes recommendations using them together with a possibly pre-existent rating data set.

We performed a live user A/B test by using these two system variants. Our results indicate that pairwise preferences are valuable and useful when the user is searching for a specific type of POI (in our case, restaurant for a specific occasion) and that, by incorporating pairwise preferences, the system is able to capture user preferences effectively and to produce better recommendations than a state of the art rating-only based solution. Our results also suggest that such type of preferences can improve the usability of the recommender system, that asking users to compare items make them feel happier to use the system, and doing that the system is able to collect more preferences (pairwise comparisons vs. ratings). Moreover, our results also indicate a dependency between the user's personality, and the system usability and the user satisfaction for the preference elicitation procedure.

The rest of this paper is structured as follows. In the next section we discuss the state of the art. Next we illustrate the system-user interaction with the mobile app that we used for testing our research hypothesis. Then, we explain the implemented algorithm used for eliciting pairwise-preferences and we describe the recommendation technique for computing ranking of items. This is followed by the description of evaluation strategy used in our experiments and a comprehensive discussion of the obtained results. Finally, we formulate our conclusions and discuss future work.

2 RELATED WORK

2.1 Pairwise Preferences

Pairwise preferences (relative preferences or comparisons) are widely used even outside the RS research area, e.g., in decision making and marketing. For instance, popular techniques, such as, Analytic Hierarchy Processing (AHP) [34] or Conjoint Analysis [37], use relative comparison between product features to understand or elicit users interest. In Behavioral Economics, according to Preference Reversal theory [40], user's preferences while evaluating items individually (Separate Evaluation) differ from the preferences expressed while comparing items together (Joint Evaluation). Users tend to express preferences differently in these two situations [1] and therefore it is important for RSs to analyse the quality of recommendations generated by alternative preference models (e.g., ratings vs. pairwise preferences). Moreover, pairwise comparison are shown to be useful in obtaining reliable teacher assessments [19] and peer assessment of undergraduate studies [21].

In previous research, it has been already shown that pairwise preference judgments are also faster than ratings to express, and users prefer to provide their preferences in this form [12, 22]. Additionally, the authors of [22] showed that pairwise preferences are more stable than ratings. More in general, some research works in RSs, instead of modeling user preferences using ratings, tried modeling user preferences in other forms such as group based elicitation approach using groups of items [32, 39] and choice based preference elicitation [18]. These approaches have shown to yield good performance in terms of reducing user effort, time and improving user satisfaction when compared to ratings.

Focussing on recommendation algorithms, while there are numerous RSs that generate recommendations by using user preferences in the form of ratings, recently, few authors developed recommendation techniques that combine ratings and pairwise preferences [11, 13, 14, 16, 25]. In [24] the authors also exploited feature preferences as an additional source of information to improve cold recommendation for a pairwise preferences based RSs. However, we note that these RSs do not couple the proposed recommendation algorithm with an appropriate GUI for supporting the full interaction of the user with the system. In our work, we have developed a specific interaction model, GUI components, an active learning strategy and a recommendation technique to support the full interaction of the user with the system and in addition, we have identified important conditions influencing the elicitation and effectiveness of RSs based on pairwise preferences.

2.2 Active Learning

Many researchers have proposed to use active learning strategies to enable the RS to actively decide what (preference) data should be acquired before starting the learning phase and consequently improve the system performance [33]. For instance, one of the most common approaches to address the new user problem in RSs is to present users with a selection of items to rate which have been on purpose identified by an active learning algorithm. For that reason, numerous active learning strategies were proposed for rating based RSs [15]. Active learning strategies for pairwise preference based RSs is a topic that has not been explored yet. In this paper, we propose a novel active learning strategy to elicit pairwise preferences and we describe its implementation in a mobile app. Moreover, we show that ratings can be optimally combined with pairwise preferences to compute effective recommendations. Hence this paper starts a novel research topic, i.e., active learning algorithms for RS using mixed preferences data (rating and pairwise preferences) or multiple preference data.

2.3 User Interfaces

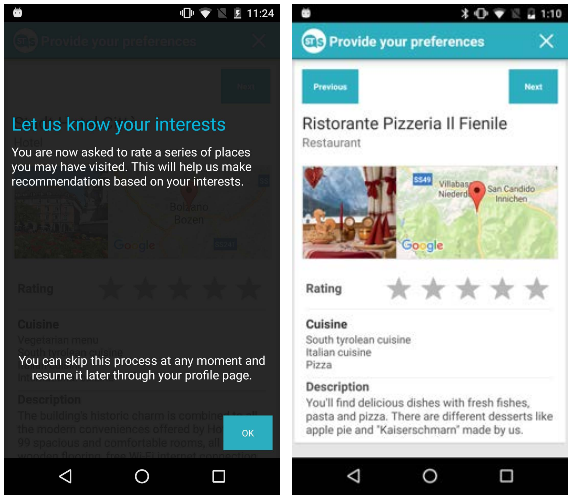
Preference elicitation requires GUIs to present item or pairs of items for users to give their preferences. Most of the current interfaces for evaluating items are based on ratings, e.g., 5 stars or thumb up/down. Some researchers proposed solutions for improving rating interfaces and better support the rating process. In [29] the authors suggested to include personalized tags and exemplars to relate rating decisions to prior ones. So, while rating interfaces are rather popular and widely used today, there has been little research work done toward pairwise preference elicitation interfaces and building pairwise preference based RS [5, 22, 30]. As a consequence, the recent research in the field of RS using pairwise preferences just considered pairwise scores, which measure how much an item is preferred to another, obtained by transforming ratings to pairwise score, i.e., by taking the difference of the ratings. However, a true pair scores acquisition interface needs a GUI that effectively enables users to compare items and enter to what extent one is preferred to the other. In [5] the authors presented a GUI for a movie recommender that uses sliders: the closer the slider pointer is dragged to an item, the more this item is preferred to the other. In this work, besides building a recommendation model based on the combination of ratings and pairwise preferences, we also develop a novel user interface for eliciting binary pair scores that is adapted to the mobile scenario.

3 PAIRWISE-BASED RECOMMENDER

In order to test the hypotheses listed in the introductory section, we have used a fully functional context-aware RS mobile app, namely, STS[1]. STS has been successfully used as testing prototype for many recommendation approaches [6–9, 28]. STS is publicly available on Google play store[2]. It enables the user to register, enter preferences in the form of ratings and obtain context-aware recommendations for POIs.

We have implemented two RS variants using the core STS app. The first variant, which we call STS-R, is our baseline and it is the pre-existent application. STS-R uses only ratings to compute recommendations. Figure 1 shows the rating interface used in STS-R. In STS-R, the implemented active learning strategy, which identifies POIs that are presented to the user to rate, is called "binary prediction" [8]. The recommendation algorithm used in STS is based on context-aware matrix factorization (CAMF) [2]. The model generates recommendations using the user assessed personality, the user's age and gender (if available), contextual information, such as, weather, feeling, travel goal or temperature, and the full set of available ratings. We note that a new user of STS, even though he has not entered any ratings, can still obtain recommendations. These are generated by the CAMF model on the base of other user data, namely the user personality. User personality is acquired by asking the user to complete the Five-Item Personality Inventory (FIPI) [17]. Details about the STS-R active learning strategy and its recommendation algorithm can be found in [7].

The second variant is named STS-RC. It extends STS-R by enabling the user to enter both ratings and pairwise preferences. Hence, STS-RC differs from STS-R only: a) in an additional GUI, which lets the user to compare items identified by a novel active learning procedure, and b) in the recommendation algorithm that generates personalised rankings of the recommended POIs by exploiting both ratings and pairwise preferences (pairwise scores).



Let us know your interests

You are now asked to rate a series of places you may have visited. This will help us make recommendations based on your interests.

You can skip this process at any moment and resume it later through your profile page.

Ristorante Pizzeria Il Fienile

Restaurant

Rating ★ ★ ★ ★ ★

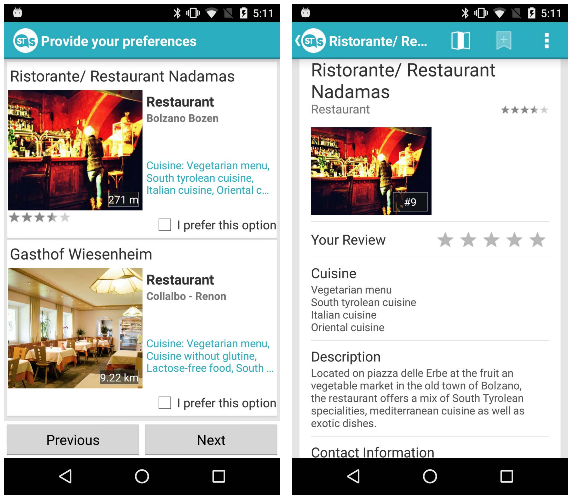
Cuisine

South tyrolean cuisine
Italian cuisine
Pizza

Description

You'll find delicious dishes with fresh fishes, pasta and pizza. There are different desserts like apple pie and "Kaiserschmarn" made by us.

Figure 1: Rating Interface



Ristorante/ Restaurant Nadamas

Restaurant
Bolzano Bozen

Cuisine: Vegetarian menu, South tyrolean cuisine, Italian cuisine, Oriental c...

★★★★☆

□ I prefer this option

Gasthof Wiesenheim

Restaurant
Collalbo - Renon

Cuisine: Vegetarian menu, Cuisine without glutine, Lactose-free food, South ...

□ I prefer this option

Ristorante/ Restaurant Nadamas

Restaurant ★★★★☆

Your Review ★ ★ ★ ★ ★

Cuisine

Vegetarian menu
South tyrolean cuisine
Italian cuisine
Oriental cuisine

Description

Located on piazza delle Erbe at the fruit an vegetable market in the old town of Bolzano, the restaurant offers a mix of South Tyrolean specialities, mediterranean cuisine as well as exotic dishes.

Contact Information

Figure 2: Pairwise Preference interface

[1] http://rerex.inf.unibz.it/

[2] https://play.google.com/store/apps/details?id=it.unibz.sts.android

3.1 STS-RC Preference Acquisition

Eliciting pairwise preferences in a mobile environment requires a GUI that presents pair of items to the users and enables them to compare these pairs. Figure 2, left, shows the actual GUIs that we have designed and used in our experiment. The user can enter pairwise preferences by checking 'I prefer this option' to signal the preferred POI among the two that are presented. Moreover, the user can click on a POI image to browse the POI details (see Figure 2, right) and she can also rate the POI if she wishes to do so.

In order to identify the item pairs to present to the user for comparison, we have implemented an active learning procedure that satisfies the following requirements:

- In order to simplify the pairwise comparison, compared items must be somehow similar (comparable).
- At least one of the items to be compared must be predicted as highly relevant to the user by the recommendation algorithm so that the user will find more plausible and easier to compare it to another.
- In order to improve the RS accuracy, the selected pairs of items to compare must express user preference information about relevant items.

In order to better understand the proposed active learning procedure for eliciting pairwise preferences, let us assume that a tourist has registered with STS-RC and the system has generated a personalized recommendation list on the suggestion screen using the CAMF algorithm described above. Using the generated ranked list R, we propose the user to compare some of the top recommended items with other items that are similar to them and are also predicted as top relevant items (higher position in the ranked list). Ultimately, for each top item i in the ranked list R, we identify the item j_i in the list R that maximizes the Equation 1. Then, we form the pairs (i, j_i) and we present them to the user in the same order of the items i in the ranked list R.

$$j_i = argmax_j\{(V(i) \cdot V(j)) * \frac{1}{R(j)}\} \tag{1}$$

Hereby, $R(i)$ is the rank position of the item i in the recommendation list, $V(i) = (v_1, \ldots, v_n)$ denotes the n dimensional Boolean feature vector that represents the item i and n is the number of features. For example, if $V(i) = (0, 1, 0, 1)$ this means that item i possesses the second and the fourth features, but not the first and the third. Previous research has shown that representing an item by a fixed number of n Boolean features is useful in Tourism [4, 10, 28]. In order to build the vector representation of the available items we have used various sources of information including item categories, such as, "museum" or "hotel", and keywords extracted from the item name. All the item's features are obtained through a web-service provided by the Regional Association of South Tyrol's Tourism Organizations[3]. After removing redundancy, each item is represented by 145 features. We note that the above preference acquisition technique requires a recommendation algorithm for ranking items. In our application, as we mentioned above, we used the CAMF algorithm which is implemented in STS.

3.2 STS-RC Pairwise Preferences Prediction and Item Ranking

In [25] we proposed a user-based Nearest-Neighbor (NN) approach for predicting unknown pairwise preferences. It uses Goodman and Kruskal's gamma (GK) as user-to-user similarity measure [25]. GK is calculated as $\frac{P-Q}{P+Q}$, where P is the number of item pairs for which the two users have expressed the same preference (concordant pairs) and Q is the number of item pairs for which the two users have different preferences (discordant pairs). We note that GK does not exploit contextual information while calculating similarity among two users. Therefore, in this work, we have modified and adapted GK so that it can also take into account pairwise preference given under contextual situations. For instance, a contextual pairwise preference may express: "I prefer item i over item j when traveling with family".

Let P_u be the set of pairwise preferences given by a user u without any reference to a specific context and P_u^c be the set of pairwise preferences given by the user u when comparing items in a particular contextual condition, e.g., when the user is searching for a POI to visit in a "business trip" or when "traveling with family". In our experiments, users have provided both types of preferences, and in order to appropriately use both of them we have modified the calculation of the user-to-user GK similarity as follows: if the pairwise preference, which is compared in the two users' profiles, is from P_u^c and it is referring to the contextual situation of the user when is requesting a recommendation, we give more importance to it by weighing it more. In our experiments we weigh it twice, i.e., we add 2 to P (Q) if it is (dis)concordant in the two users profiles. For instance, if the user is looking for recommendations for a birthday party context, we weigh twice all the pairwise preferences concordances and discordances, which are assessed on comparisons made under this contextual situation. The other pairwise preferences concordances and discordances are weighed one, as usual.

Finally, the NN prediction formula for unknown pairwise score prediction is:

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

where r_{uij} is the true pairwise score given by the user u to the pair (i, j) with possible values +1 or -1, and r^*_{uij} is the predicted value, when the user has not expressed that comparison. U_{ij} is the set of users that have compared the items i and j, and $sim(u, v)$ is the similarity score, which is computed here with GK.

After predicting the missing pairwise scores, i.e., those not expressed by the user, we aggregate them to compute a personalized item score v_{ui} by averaging the r^*_{uij} predictions, as done by [5, 24, 25]:

$$v_{ui} \quad = \quad \frac{\sum_{j \in I \setminus \{i\}} r^*_{uij}}{|I|} \tag{3}$$

Note that, if r_{uij} is known, then in the above formula the prediction r^*_{uij} is replaced with the true value r_{uij}. Finally, for each user u the items are then ranked and recommended by descending values of the v_{ui} scores.

[3] http://www.lts.it.

4 EXPERIMENTAL STRATEGY

We used STS-R and STS-RC to validate our research hypotheses, given in the introductory section. We performed a live user study, as an A/B test (between group). The initial data set of STS ratings[4] is common for both systems and contains 2534 ratings given by 325 users. In STS-RC these ratings are used to generate an initial set of pairwise scores (ratings differences). In fact, as we have already mentioned, one can easily consider two ratings of the same user, r_{ui} and r_{uj}, and, if they have a different value, convert them into a pairwise score as $r_{uij} = sgn(r_{ui} - r_{uj})$, where $sgn()$ is the sign function, which maps positive values to +1 and negative values to -1. Both systems are bootstrapped with the preference information contained in this data set and then augmented with the additional preferences that were entered by the users in the two variants during the experiment.

In order to understand the effectiveness of modeling user preferences in the form of pairwise preferences for situations where a user has a specific goal and is looking for a specific type of item recommendations, we conducted a between-group study design, comparing two preference elicitation approaches, i.e., ratings (in STS-R) or pairwise preferences and ratings (in STS-RC), and the recommendation techniques using those preferences. The evaluation strategy comprises the following stages and steps (the same for both groups):

- **Phase I:** Initial preference elicitation without any specific goal.
  - User preference elicitation;
  - User assessment of the system usability and the recommendation quality using a questionnaire.
- **Phase II:** Preference elicitation and recommendations for a user with a specific goal.
  - Recommendation matching the user's specific goal;
  - User input of additional preferences to better understand the user needs and improve the recommendations;
  - User assessment of the preference elicitation procedure and the recommendation quality.

The above steps are further described in the following. A user is randomly assigned to either STS-R or STS-RC so as to provide his/her preferences and receive recommendations. 58 users registered for the experiment (29 used STS-R and 29 used STS-RC). They are aged between 20 to 50 (the mean is 28), 17% females and 55% males. The majority were either undergraduates, PhD students or working employees. The experiment participants thus are among the typical users of tourism apps.

The registration process is the same for both systems: each user enters his name, password, birthday, gender and fills out the FIPI questionnaire. After that, each user receives immediately some general recommendations, for several different types of POIs (e.g., restaurants, events, museums, etc.) that are predicted using the CAMF algorithm. Then, in an initial preference elicitation stage (Phase I), the user preferences are gathered in the form of either ratings (in STS-R) or pairwise preferences and ratings (in STS-RC), depending on the system to which the user was assigned. Items to rate or item pairs to compare are identified using the active learning strategy described in Section 3. We did not request the users to provide any precise number of preferences: they were free to enter as many preferences as they liked. After providing these preferences, the users were presented with additional recommendations. In the second step of phase I, users were asked to evaluate the system usability and the perceived recommendation quality using questionnaires. We adopted questionnaires used in similar experiments [5, 28]. Specifically, to evaluate the system usability, we used the System Usability Scale (SUS) questionnaire [3]. SUS is one of the most popular post-study standardized questionnaires and it also allows to measure the perceived system usability with a small sample population (i.e., 8–12 users) [36]. SUS contains 10 statements, half positive and half negative, shown in Table 1. Users judge each statement with a five-point Likert scale, ranging from strongly disagree (1) to strongly agree (5) with the statement.

Table 1: SUS statements concerning the perceived system usability

1	I think that I would like to use this system frequently.
2	I found the system unnecessarily complex.
3	I thought the system was easy to use.
4	I think that I would need the support of a technical person to be able to use this system.
5	I found the various functions in the system were well integrated.
6	I thought there was too much inconsistency in this system.
7	I would imagine that most people would learn to use this system very quickly.
8	I found the system very cumbersome to use.
9	I felt very confident using the system.
10	I needed to learn a lot of things before I could get going with this system.

For scoring a SUS we followed the recommended procedure for calculating the overall SUS score, which ranges between 0 and 100 [35]. Several benchmarks for the SUS across different systems have been published and the average SUS score computed in a popular benchmark of 500 studies is 67 [35]. In our experiments, we used this value as benchmark for our system's usability.

The perceived recommendation quality was instead assessed by using the related six statements of the questionnaire presented in [26] and shown in Table 2. Statements are judged like in SUS and a cumulative score is computed in a similar manner.

In Phase II of the experiment, each user was given a scenario description in order to have a specific goal (objective) oriented situation and was asked to complete the task described in the scenario. The exact user task scenario is as follows. The user is supposed to have an evening off to take his colleagues to a restaurant for his birthday party in South Tyrol. The user is asked to use the system to find a suitable restaurant for this event. First the user is requested to specify this event (goal) in the STS's settings. The user is then presented with restaurant recommendations which are predicted as

[4] https://www.researchgate.net/publication/305682479_Context-Aware_Dataset_STS_-_South_Tyrol_Suggests_Mobile_App_Data.

Table 2: Questionnaire items concerning the perceived recommendation quality.

1	I liked the items suggested by the system.
2	The suggested items fitted my preference.
3	The suggested items were well-chosen.
4	The suggested items were relevant.
5	The system suggested too many bad items.
6	I didn't like any of the suggested items.

suitable for the event (step 1 of phase II). Next, in order to better understand the user needs and to provide better recommendations, the user is then asked to enter some more preferences on the searched items (restaurants) by focusing on his specific information needs (step 2 of phase II). Finally, the user is presented with an improved set of recommendations based on all the entered preferences. The user is supposed to browse these recommendations and to select a restaurant from the suggestion list and to bookmark it.

Finally, in step three of the Phase II, users were asked to fill a survey on the user satisfaction with the preference elicitation (PE) procedure and on the perceived recommendation quality. The statements that we used to evaluate the preference elicitation process [26] are shown in Table 3. Statements are judged like in SUS and a cumulative score is computed in a similar manner.

Table 3: Questionnaire items concerning the user satisfaction with the preference elicitation procedure.

1	I have fun using the system.
2	Using the system is a pleasant experience.
3	The system improved the quality of recommendation when additional preferences are added.
4	The system makes me more aware of my choice options.
5	I feel bored when I am using the system.

5 RESULTS

Tables 4 and 5 show the results of the questionnaires concerning the perceived system usability (see Table 1), the recommendation quality (see Table 2), the satisfaction with the preference elicitation process (see Table 3), the mean reciprocal rank of the chosen POI[5] and the number of preferences collected on average per user (ratings in STS-R or pairwise preferences in STS-RC). We will discuss each of them in the following subsections.

5.1 System Usability

Analyzing the results of the SUS questionnaire concerning the perceived system usability (second step of Phase I), we found that the SUS score of STS-RC is higher than the SUS score of STS-R, 76.29 vs. 71.93, and the SUS score of both variants is well above the benchmark of 67. This difference between the SUS score of STS-RC and STS-R is significant (Mann-Whitney test, p = 0.048). We also analyzed the user replies to each individual questionnaire statements. In particular, we found that STS-RC is perceived to be significantly less complex than STS-R is: the user's agreement with statement 2 of SUS, assessing the perceived complexity of the system, is significantly different between STS-RC and STS-R ($p < 0.01$). Overall, the results in this experiment support our first research hypothesis (H_1): pairwise preferences can be effectively used in mobile apps.

Table 4: Results for various metrics and percentage of improvement of STS-RC over STS-R (an asterisk means that STR-RC is significantly better than STR-R, p<0.05).

		STS-R	STS-RC	RC-R %
SUS score		71.93	76.29*	6%
PE score		54.96	60.24	9.6%
MRR (Phase II)		0.210	0.271*	29%
Collected Preferences (on average per user)	Phase I	9.37	13.44	43%
	Phase II	12.65	15.24	20%

5.2 User Satisfaction for the Preferences Elicitation Process

As we explained above, in the second phase of the experiment, the user task is focused on searching a restaurant and the additional preferences inserted by the user are related to restaurants. We have measured the user satisfaction for the preferences elicitation (PE) procedure to understand which one is more effective in this situation.

Analyzing the replies of the survey, we found that the STS-RC and STS-R scores are 60.24 and 54.96, respectively, even though the difference is not statistically significant (p = 0.06). Lack of statistical significance is probably due to the small number of participants. Our results, thus, only marginally support our second research hypothesis (H_2), i.e., that pairwise preferences can lead to a larger user satisfaction for the preference elicitation procedure when the user is focusing on a specific information need/recommendation.

Furthermore, when we analyzed the user replies to each individual questionnaire statements, we found that STS-RC tends to make users feel happier to use the system albeit not significantly so: for statement 1, concerning the preference elicitation, users had a larger agreement for STS-RC than for STS-R (p = 0.094). However, for statement 3, users felt that STS-RC, in comparison to STS-R, improved the quality of the recommendations when additional preferences were added, and significantly so (p = 0.035).

5.3 Perceived Recommendation Quality

Table 5 shows the results of perceived recommendation quality at the end of the first and second phase. Analysing the replies to the questionnaire at the end of the first phase, we found that the STS-RC and STS-R scores are 58.14 and 63.63 (p= 0.19), respectively, but not significantly different. This means that when there is not a specific goal for the recommendation, STS-R may be able to generate better recommendations compared to STS-RC but the claim needs to be supported by further experiments.

[5] The exact definition of this metric is reported later in the document.

Table 5: Perceived recommendation quality results (a double asterisk means that STR-RC is significantly better in the second phase compared to the first one, p<0.01).

	Phase I	Phase II	(Phase II - Phase I) %
STS-R	63.63	64.52	1.4%
STS-RC	58.14	65.51**	12.6%

Hence, our results suggest that pairwise preferences are not particularly effective in situations when the user does not have a specific goal. However, in the second phase of the study, i.e., when the users are focusing on a particular goal and request the system to provide recommendations targeting this goal, STS-RC and STS-R have similar perceived recommendation quality: 65.51 vs. 64.52. Interestingly, we observe that STS-R has similar perceived recommendation quality in the first phase (without a particular goal) 63.63 and the second phase (with a particular goal) 64.52. Conversely, there is a significant improvement of this score for STS-RC between the first and the second phase: 58.14 vs 65.51 (Mann-Whitney test and paired t-test, $p < 0.01$). Hence, since STS-RC is significantly better in the second stage than in the first, this clearly shows that comparisons are more effective to model user preferences if the user has a specific objective and is looking for a specific type of recommendations. It is worth noting that users reported that comparing item pairs made them imagine a ranking of the compared items and they also felt that the recommendation list generated by STS-RC was appealing since it was close to their ideal ranking of the recommended items. However, we must note that there may be a carry-over effects in the second phase due to the usage of the application in the first phase; users could have benefitted from it in the second phase. Future experiments might test it with a different experiment design. However, this effect is independent from the variant used by the user, hence it is not influencing the observation that STS-RC improved more the recommendation quality than STS-R when the recommendation goal is more specific.

Moreover, we recall that we have asked each user, at the end of the second phase, to bookmark one suggestion that they believe it fits their preferences. We then measured the ranking error of the system using mean reciprocal rank (MRR) [38] by assuming that this bookmarked item is the relevant one. MRR computes, for each user, the inverse of the rank position in the recommended list, of the bookmarked item (higher MRR values are better). We found that the MRR of STS-RC is 0.271 and it is statistically significantly higher than the MRR of STS-R, which is 0.210 ($p = 0.039$). These results support the third hypothesis (H_3) that we made, i.e., preferences elicited in the form of pairwise preferences improve more the recommendation accuracy and quality, when there is a clear objective for the recommendations.

Finally, we also looked at the number of preferences entered in the two variants in the two experimental phases. In the first one, on average per user, 9.37 ratings and 13.44 pairwise preferences were provided in STS-R and STS-RC, respectively. In the second phase, on average per user, 12.65 ratings and 15.24 pairwise preferences[6] were provided by the users using STS-R and STS-RC, respectively. Thus, both systems collected more preferences in the second phase compared to first phase. We also observe that users entered more preferences by using STS-RC than using STS-R in both phases of the experiment, albeit the difference is not statistically significant.

5.4 Impact of User Personality

Finally, we wanted to understand if there exists any dependency between the user's personality with the perceived system usability and the user evaluation for the preference elicitation procedure.

We recall that each user, when registered to the system (both STS-R and STS-RC), filled in the FIPI questionnaire, used to assess the big five personality traits: openness, conscientiousness, extroversion, agreeableness, neuroticism. Users replied on a seven-points Likert scale ranging from "strongly disagree" (1) to "strongly agree"(7).

In previous research, it was shown that there exists a relation between user personality and: preference elicitation methods [20], online user behavior and social networking [27]. We, therefore, tried to identify if there are any dependencies between user personality traits and SUS scores or PE scores. Tables 6 and 7 show the average scores of PE and SUS of users with low and high personality traits in the STS-R and STS-RC systems. We used a median split to divide users into the low (users with personality trait score less than the median) and high personality trait groups (users with personality trait score equal to or greater than the median). We then double checked if there is any significant difference of SUS scores or PE scores between low and high personality trait groups.

Observing Tables 6 and 7, we can state that users who possess high extroversion tend to like the rating-based elicitation procedure more than the users who possess low extroversion ($p = 0.043$). This means that high extroverts, who are more socially active, energetic and talkative, like using ratings more to express their preferences when compared to users who possess low extroversion. High extroverts tend to be good at judging and making the decisions [41] and, possibly, this assertive behavior makes them comfortable to make rating evaluations more than those that are not extroverts. We can also observe that users who possess high agreeableness tend to like the pairwise preference based preference elicitation mechanism more than users who possess low agreeableness ($p = 0.016$). Moreover, users who possess high agreeableness tend to give significantly higher usability scores compared to users who possess low agreeableness ($p = 0.027$). This means that high agreeable users, who are kind, sympathetic, and warm in nature, express their preferences using pairwise preferences and use more pairwise preferences based RS when compared to users who possess low agreeableness. We also performed correlation analysis between the user's personality of both systems and SUS score or PE scores. We found that among STS-RC users agreeableness has a significantly positive correlation with perceived system usability (Spearman's rank correlation coefficient = 0.413, $p=0.02$) and preference elicitation procedure (Spearman's rank correlation coefficient = 0.429, $p=0.026$). This result further supports the previous observations made for pairwise comparisons and personality.

6 CONCLUSIONS

In this work we aimed at understanding when eliciting pairwise preferences is meaningful and beneficial and, more in general, at

[6] In addition to pairwise preferences users of STS-RC entered very few ratings, 28.

Table 6: Comparison of PE scores between the two groups of users having low and high scores for the five FIPI personality traits (an asterisk means p < 0.05). The two systems, STS-R and STS-RC, are considered separately.

	STS-RC		STS-R	
	Low value	**High value**	**Low value**	**High value**
Openness	61.78	58.82	49.99	53.74
Conscientiousness	57.46	61.39	51.33	56.53
Extraversion	56.86	61.59	48.19	57.84*
Agreeableness	50.58	67.97*	51.47	55.74
Neuroticism	61.49	58.49	53.43	55.36

Table 7: Comparison of SUS scores between the two groups of users having low and high scores for the five FIPI personality traits (an asterisk means p < 0.05). The two systems, STS-R and STS-RC, are considered separately.

	STS-RC		STS-R	
	Low value	**High value**	**Low value**	**High value**
Openness	75.62	72.72	71.25	68.52
Conscientiousness	77.50	74.84	70.00	72.63
Extraversion	74.37	77.21	71.13	71.94
Agreeableness	71.25	79.55*	72.18	69.18
Neuroticism	63.87	62.86	71.25	71.91

providing an empirical confirmation that RSs based on pairwise preferences offer a valid alternative to traditional systems based on ratings.

By conducting an A/B test, we have shown that, when the user is searching for a specific recommendation, RSs with pairwise preferences can perform equally or better than state of the art rating based solutions. Overall, our results confirm our main research hypothesis: pairwise preferences are a viable approach to preference elicitation in RSs; especially in situations and scenarios where a user has a clear objective, a system using pairwise preferences can offer a better recommendation experience to the user.

In particular, we have shown that pairwise preferences can lead to higher usability of the RS and also to a larger user satisfaction for the preference elicitation procedure, when the user is focusing on a specific information need and, as a consequence, the choice set to make the decision is small. Additionally, we have shown that pairwise preferences make users feel happier to use the RS when compared to ratings, and the RS using pairwise preferences is able to collect more such preferences than ratings. We also have shown that there is a dependency between the user's personality, and the system usability and the user evaluation for the preference elicitation process. This suggests that users' personality can be useful to decide which type of preference elicitation may be used, and by doing so, one can collect more preferences, which in turn can lead to better recommendations and higher usability of the RS.

We stress that we have obtained these results by developing novel techniques to effectively implement an RS that combines pairwise preferences and ratings: a novel active learning strategy for pairwise preference elicitation, novel ranking algorithm and a specifically designed GUI.

We also acknowledge limitations of the presented work. First of all, we note that we have considered one single example of focused search, i.e., looking for a specific type of item recommendations (restaurant). In fact, many of such situations can be identified during a user interaction with the RS in his daily usage. Moreover, we have designed two systems that either request the user to rate or to compare. But the two approaches may be mixed. For instance, imagine that an item is recommended to the user and the user has experienced it. Then in order to get additional preference knowledge from the user, it may be useful to understand if rating this item is effective or it would be better to ask the user to compare it to previously experienced similar ones.

Our results also suggest other future lines of investigation. For instance, an analysis of users' personalities and whether they are goal directed in their search could help decide on the related recommendation mechanism, e.g., this could be based on pair-wise preferences if users are highly agreeable or goal-directed. Future work will also try extending our findings concerning personality traits and preferences for recommendation mechanisms.

Finally, we plan to explore session data and use them in our models. In fact, the pairwise preferences, which are elicited in the presence of a user goal, may contain information that can be useful for a short time, i.e., session based recommendations. In that respect, further work is needed to deepen how to explore session data and incorporate them into our models and improve recommendations.

REFERENCES

[1] Dan Ariely. 2009. *Predictably irrational.* HarperCollins New York.

[2] Linas Baltrunas, Bernd Ludwig, Stefan Peer, and Francesco Ricci. 2012. Context relevance assessment and exploitation in mobile recommender systems. *Personal and Ubiquitous Computing* 16, 5 (2012), 507–526.

[3] Aaron Bangor, Philip T Kortum, and James T Miller. 2008. An empirical evaluation of the system usability scale. *Intl. Journal of Human–Computer Interaction* 24, 6 (2008), 574–594.

[4] Henry Blanco and Francesco Ricci. 2013. Acquiring user profiles from implicit feedback in a conversational recommender system. In *Proceedings of the 7th ACM conference on Recommender systems.* ACM, 307–310.

[5] 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.

[6] Matthias Braunhofer, Mehdi Elahi, and Francesco Ricci. 2014. STS: A Context-Aware Mobile Recommender System for Places of Interest.. In *UMAP Workshops.*

[7] Matthias Braunhofer, Mehdi Elahi, and Francesco Ricci. 2014. Usability assessment of a context-aware and personality-based mobile recommender system. In *International conference on electronic commerce and web technologies.* Springer, 77–88.

[8] Matthias Braunhofer, Mehdi Elahi, and Francesco Ricci. 2015. User personality and the new user problem in a context-aware point of interest recommender system. In *Information and Communication Technologies in Tourism 2015.* Springer, 537–549.

[9] Matthias Braunhofer, Mehdi Elahi, Francesco Ricci, and Thomas Schievenin. 2013. Context-aware points of interest suggestion with dynamic weather data management. In *Information and communication technologies in tourism 2014.* Springer, 87–100.

[10] Derek Bridge and Francesco Ricci. 2007. Supporting product selection with query editing recommendations. In *Proceedings of the 2007 ACM conference on Recommender systems.* ACM, 65–72.

[11] Armelle Brun, Ahmad Hamad, Olivier Buffet, and Anne Boyer. 2010. Towards preference relations in recommender systems. In *Workshop on Preference Learning, European Conference on Machine Learning and Principle and Practice of Knowledge Discovery in Databases (ECML-PKDD 2010).* Citeseer.

[12] Ben Carterette, Paul N Bennett, David Maxwell Chickering, and Susan T Dumais. 2008. Here or there: Preference judgments for relevance. In *Advances in Information Retrieval 2008.* Springer, 16–27.

[13] Maunendra Sankar Desarkar and Sudeshna Sarkar. 2012. Rating prediction using preference relations based matrix factorization.. In *UMAP Workshops.* Citeseer.

[14] Maunendra Sankar Desarkar, Roopam Saxena, and Sudeshna Sarkar. 2012. Preference relation based matrix factorization for recommender systems. In *International conference on user modeling, adaptation, and personalization.* Springer, 63–75.

[15] Mehdi Elahi, Francesco Ricci, and Neil Rubens. 2016. A survey of active learning in collaborative filtering recommender systems. *Computer Science Review* 20 (2016), 29–50.

[16] David F Gleich and Lek-heng Lim. 2011. Rank aggregation via nuclear norm minimization. In *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining.* ACM, 60–68.

[17] Samuel D Gosling, Peter J Rentfrow, and William B Swann. 2003. A very brief measure of the Big-Five personality domains. *Journal of Research in personality* 37, 6 (2003), 504–528.

[18] Mark P Graus and Martijn C Willemsen. 2015. Improving the User Experience during Cold Start through Choice-Based Preference Elicitation. In *Proceedings of the 9th ACM Conference on Recommender Systems.* ACM, 273–276.

[19] Sandra Heldsinger and Stephen Humphry. 2010. Using the method of pairwise comparison to obtain reliable teacher assessments. *The Australian Educational Researcher* 37, 2 (2010), 1–19.

[20] Rong Hu and Pearl Pu. 2009. A comparative user study on rating vs. personality quiz based preference elicitation methods. In *Proceedings of the 14th international conference on Intelligent user interfaces.* ACM, 367–372.

[21] Ian Jones and Lara Alcock. 2014. Peer assessment without assessment criteria. *Studies in Higher Education* 39, 10 (2014), 1774–1787.

[22] Nicolas Jones, Armelle Brun, and Anne Boyer. 2011. Comparisons instead of ratings: Towards more stable preferences. In *Web Intelligence and Intelligent Agent Technology (WI-IAT), 2011 IEEE/WIC/ACM International Conference on*, Vol. 1. IEEE, 451–456.

[23] Saikishore Kalloori. 2017. Pairwise Preferences and Recommender Systems. In *Proceedings of the 22nd International Conference on Intelligent User Interfaces Companion.* ACM, 169–172.

[24] 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, 289–293.

[25] 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.

[26] Bart P Knijnenburg, Martijn C Willemsen, Zeno Gantner, Hakan Soncu, and Chris Newell. 2012. Explaining the user experience of recommender systems. *User Modeling and User-Adapted Interaction* 22, 4-5 (2012), 441–504.

[27] Michal Kosinski, Yoram Bachrach, Pushmeet Kohli, David Stillwell, and Thore Graepel. 2014. Manifestations of user personality in website choice and behaviour on online social networks. *Machine learning* 95, 3 (2014), 357–380.

[28] Thuy Ngoc Nguyen and Francesco Ricci. 2017. A Chat-Based Group Recommender System for Tourism. In *Information and Communication Technologies in Tourism 2017.* Springer, 17–30.

[29] Tien T Nguyen, Daniel Kluver, Ting-Yu Wang, Pik-Mai Hui, Michael D Ekstrand, Martijn C Willemsen, and John Riedl. 2013. Rating support interfaces to improve user experience and recommender accuracy. In *Proceedings of the 7th ACM conference on Recommender systems.* ACM, 149–156.

[30] Syavash Nobarany, Louise Oram, Vasanth Kumar Rajendran, Chi-Hsiang Chen, Joanna McGrenere, and Tamara Munzner. 2012. 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.* ACM, 2035–2044.

[31] Francesco Ricci, Lior Rokach, and Bracha Shapira. 2015. Recommender Systems: Introduction and Challenges. In *Recommender Systems Handbook.* Springer.

[32] Silvia Rossi, Francesco Barile, Sergio Di Martino, and Davide Improta. 2017. A comparison of two preference elicitation approaches for museum recommendations. *Concurrency and Computation: Practice and Experience* 29, 11 (2017).

[33] Neil Rubens, Mehdi Elahi, Masashi Sugiyama, and Dain Kaplan. 2015. Active learning in recommender systems. In *Recommender systems handbook.* Springer, 809–846.

[34] Thomas L Saaty. 2008. Decision making with the analytic hierarchy process. *International journal of services sciences* 1, 1 (2008), 83–98.

[35] Jeff Sauro. 2011. Measuring usability with the system usability scale (SUS), http://www.measuringusability.com/sus.php. (2011).

[36] Jeff Sauro and James R Lewis. 2016. *Quantifying the user experience: Practical statistics for user research.* Morgan Kaufmann.

[37] Michael Scholz. 2008. From consumer preferences towards buying decisions. *21st Bled eConference eCollaboration: Overcoming Boundaries Through Multi-Channel Interaction* (2008), 223–235.

[38] Hinrich Schütze, Christopher D Manning, and Prabhakar Raghavan. 2008. *Introduction to information retrieval.* Vol. 39. Cambridge University Press.

[39] Loren Terveen Shuo Chang, F. Maxwell Harper. 2015. Using Groups of Items for Preference Elicitation in Recommender Systems. In *CSCW '15 Proceedings of the 18th ACM Conference on Computer Supported Cooperative Work and Social Computing* (2015-03-11). ACM, 1258–1269. http://dl.acm.org/citation.cfm?id=2675210

[40] Paul Slovic and Sarah Lichtenstein. 1983. Preference reversals: A broader perspective. *The American Economic Review* 73, 4 (1983), 596–605.

[41] Joshua Wilt and William Revelle. 2009. Extraversion. (2009).