# Explicit and Implicit User Preferences in Online Dating

Joshua Akehurst, Irena Koprinska, Kalina Yacef,
Luiz Pizzato, Judy Kay, and Tomasz Rej

School of Information Technologies, University of Sydney, Australia
firstname.lastname@sydney.edu.au

**Abstract.** In this paper we study user behavior in online dating, in particular the differences between the implicit and explicit user preferences. The explicit preferences are stated by the user while the implicit preferences are inferred based on the user behavior on the website. We first show that the explicit preferences are not a good predictor of the success of user interactions. We then propose to learn the implicit preferences from both successful and unsuccessful interactions using a probabilistic machine learning method and show that the learned implicit preferences are a very good predictor of the success of user interactions. We also propose an approach that uses the explicit and implicit preferences to rank the candidates in our recommender system. The results show that the implicit ranking method is significantly more accurate than the explicit and that for a small number of recommendations it is comparable to the performance of the best method that is not based on user preferences.

**Keywords:** Explicit and implicit user preferences, online dating, recommender systems.

## 1 Introduction

Online dating websites are used by millions of people and their popularity is increasing. To find dating partners users provide information about themselves (*user profile)* and their preferred partner (*user preferences*); an example using predefined attributes is shown in Fig. 1. In this paper we focus on the user preferences, which is an important issue in behavior informatics [8]. We distinguish between *explicit* and *implicit* user preferences. The explicit user preferences are the preferences stated by the user as shown in Fig. 1. The implicit user preferences are inferred from the interactions of the user with other users.

Online dating is a new research area, with only a few published papers in the last year. Kim et al. [1] proposed a rule-based recommender that learns from user profiles and interactions. In another paper of the same group, Cai et al. [2] introduced a collaborative filtering algorithm based on user similarity in taste and attractiveness. McFee and Lanckriet [3] proposed an approach that learns distance metrics optimized for different ranking criteria, with evaluation in online dating. Diaz et al. [4] developed an approach for learning a ranking function that maximises the number of positive interactions between online dating users based on user profiles. In our previous work [5] we defined a histogram-based model of implicit preferences based

L. Cao et al. (Eds.): PAKDD 2011 Workshops, LNAI 7104, pp. 15–27, 2012.

on successful interactions, and showed that it can be used to generate better recommendations than the explicit preferences, when used in a content-based reciprocal recommendation algorithm.

In this paper we re-examine the effectiveness of explicit and implicit user preferences. There are three main differences with our preliminary exploration [5]. Firstly, in this paper we propose a different model of implicit preferences; it is learned from both successful and unsuccessful interactions, as opposed to being inferred from successful interactions only. Secondly, we evaluate the predictive power of the explicit and implicit user preferences in general, not as a part of a specific recommendation system. Thirdly, we propose a new method for using these preferences in our latest recommender, a hybrid content-collaborative system [6]. For a given user, it generates a list of candidates that are likely to have reciprocal interest and then ranks them to produce a shortlist of personalized recommendations so that users are quickly engaged in the search. In [6] the candidate ranking was done using the interaction history of the candidates, whereas in this paper we investigate the use of user preferences.

It is important to note that in the area of preference learning there has been little work on evaluating the relative power of explicit preferences, since this information is not normally available. Our work addresses this shortcoming for the case of online dating, where the explicit preferences are available.

| | **My attributes** | **Whom I am looking for** |
|---|---|---|
| **Age** | 44 years old | 35 yrs - 50 yrs old |
| **Location** | Perth, WA | within 25 km of 6030 |
| **Height** | 5'9/175 cm | at most 5'11/180 cm |
| **Body type** | Athletic | Slim, Average, Athletic |
| **Smoking** | Trying to quit | - |
| **Relationship status** | Divorced | Single, Divorced, Widowed, Separated |
| **Have children** | Yes, have children who don't live at home<br>**How many** 3<br>**Age range** 18 yrs - 23 yrs | Yes, have children who don't live at home, Yes, have children living at home, No, have no children |
| **Personality** | Average | - |
| **Eye colour** | Blue | - |
| **Hair colour** | Light Brown | - |
| **Nationality** | AUSTRALIA | - |

**Fig. 1.** User profile and explicit user preferences - example

Our contributions can be summarised as following:

- We propose a new approach for inferring the implicit user preferences. Given a target user *U*, we use *U*'s previous successful and unsuccessful interactions with other users to build a machine learning classifier that captures *U*'s preferences and is able to predict the success of future interactions between *U* and new users.
- We investigate the reliability and predictive power of the explicit and implicit user preferences for successful interactions. The results show that the explicit

preferences are not a good predictor of the success of user interaction, while the implicit preferences are a very good predictor.

- We propose an approach for using the user preferences in an existing recommender system, extending our previous work [6]. In particular, we propose to use these preferences for ranking of recommendation candidates. We compare the performance of the explicit and implicit ranking methods with another method that doesn't use user preferences.

Our evaluation was conducted using a large dataset from a major Australian dating website.

This paper is organized as follows. Section 2 describes how the users interact on the website. Section 3 defines the explicit user preferences and introduces our approach for learning the implicit user preferences. Section 4 describes the analysis of the predictive power of the explicit and implicit user preferences. Section 5 explains the proposed approach for using user preferences in a recommender system and discusses the results. Section 6 presents the conclusions and future work.

## 2 Domain Overview

We are working with a major Australian dating site. The user interaction on this site consists of four steps:

1) Creating a user profile and explicit user preferences – New users login to the web site and provide information about themselves (user profile) and their preferred dating partner (explicit user preferences) using a set of predefined attributes such as the ones shown in Fig. 1.
2) Browsing the user profiles of other users for interesting matches.
3) Mediated interaction – If a user *A* decides to contact user *B*, *A* chooses a message from a predefined list, e.g. *I'd like to get to know you, would you be interested?* We call these messages *Expressions of Interest (EOI). B* can reply with a predefined message either positively (e.g. *I'd like to know more about you.*), negatively (e.g. *I don't think we are a good match.*) or decide not to reply. When an EOI receives a positive reply, we say that the interest is *reciprocated*.
4) Unmediated interaction – *A* or *B* buy tokens from the website to send each other unmediated message. This is the only way to exchange contact details and develop further relationship.

We call an interaction between users A and B a *successful interaction* if: 1) *A* has sent an EOI to *B* and *B* has responded positively to it or if 2) *B* has sent and EOI to *A* and *A* has responded positively to it.

## 3 User Preferences

### 3.1 Explicit User Preferences

We define the explicit preferences of a user *U* as the vector of attribute values specified by *U*. The attributes and their possible values are predefined by the website.

In our study we used all attributes except *location;* for simplicity we considered only people from Sydney. More specifically, we used 19 attributes: 2 numeric (*age* and *height*) and 17 nominal (*marital status, have children, education level, occupation industry, occupation level, body type, eye color, hair color, smoker, drink, diet, ethnic background, religion, want children, politics, personality and have pets*).

In addition, and again for simplicity, we have removed all interactions between the same sex users and only compared people of opposing genders.

### 3.2 Implicit User Preferences

The implicit user preferences of a user *U* are represented by a binary classifier which captures *U*'s likes and dislikes. It is trained on *U*'s previous successful and unsuccessful interactions. The training data consists of all users *U*+ with whom *U* had successful interactions and all users *U*- with who *U* had unsuccessful interactions during a given time period. Each user from *U*+ and *U*- is one training example; it is represented as a vector of user profile attribute values and labeled as either *Success* (successful interaction with *U*) or *Failure* (unsuccessful interaction with *U*). We used the same 19 user profile attributes as the explicit user preferences listed in the previous section. Given a new instance, user $U_{new}$, the classifier predicts how successful the interaction between *U* and $U_{new}$ will be by outputting the probability for each class (*Success* or *Failure)* and assigning it to the class with higher probability.

As a classifier we employed NBTree [7] which is a hybrid classifier combining decision tree and Naïve Bayes classifiers. As in decision trees, each node of a NBTree corresponds to a test for the value of a single attribute. Unlike decision trees, the leaves of a NBTree are Naïve Bayes classifiers instead of class labels. We chose NBTree for two reasons. First, given a new instance, it outputs a probability for each class; we needed a probabilistic classifier as we use the probabilities for the ranking of the recommendation candidates, see Section 5. Second, NBTree was shown to be more accurate than both decision trees and Naïve Bayes, while preserving the interpretability of the two classifiers, i.e. providing an easy to understand output which can be presented to the user [7].

## 4 Are the User Preferences Good Predictors of the Success of User Interactions?

We investigate the predictive power of the explicit and implicit user preferences in predicting the success of an interaction between two users.

### 4.1 Explicit User Preferences

**Data**

To evaluate the predictive power of the explicit preferences we consider users who have sent or received at least 1 EOI during a one-month period (March 2010). We further restrict this subset to users who reside in Sydney to simplify the dataset. These two requirements are satisfied by 8,012 users (called *target users*) who had 115,868 interactions, of which 46,607 (40%) were successful and 69,621 (60%) were unsuccessful. Each target user $U$ has a set of *interacted users* $U_{int}$, consisting of the users $U$ had interacted with.

## Method

We compare the explicit preferences of each target user $U$ with the profile of the users in $U_{int}$ by calculating the number of matching and non-matching attributes.

In the explicit preferences the user is able to specify multiple values for a single nominal attribute and ranges for numeric attributes. For a numeric attribute, $U_{int}$ matches $U$'s preferences if $U_{int}$'s value falls within $U$'s range or $U_{int}$ has not specified a value. For a nominal attribute, $U_{int}$ matches $U$'s preferences if $U_{int}$'s value has been included in the set of values specified by $U$ or $U_{int}$ has not specified a value. An attribute is not considered if $U$ has not specified a value for it. The preferences of $U_{int}$ match the profile of $U$ if all attributes match; otherwise, they don't match.

## Results

The results are shown in Table 1. They show that 59.40% of all interactions occur between users with non-matching preferences and profiles. A further examination of the successful and unsuccessful interactions shows that:

- In 61.86% of all successful interactions $U$'s explicit preferences did not match $U_{int}$'s profile.
- In 42.25% of all unsuccessful interactions $U$'s explicit preferences matched the $U_{int}$'s profile.

Suppose that we use the matching of the user profiles and preferences to try to predict if an interaction between two users will be successful or not successful (if the profile and preferences match -> successful interaction; if the profile and preferences don't match -> unsuccessful interaction). The accuracy will be 49.43% (17,775+39,998 /115,868), and it is lower than the baseline accuracy of always predicting the majority class (ZeroR baseline) which is 59.78%. A closer examination of the misclassifications shows that the proportion of false positives is higher than the proportion of false negatives, although the absolute numbers are very similar.

In summary, the results show that the explicit preferences are not a good predictor of the success of interaction between users. This is consistent with [4] and [5].

**Table 1.** Explicit preferences - results

| | ***U*'s explicit preferences and $U_{int}$'s profile matched** | ***U*'s explicit preferences and $U_{int}$'s profile did not match** | **Total** |
|---|---|---|---|
| **Successful interactions** | 17,775 (38.14%) | 28,832 (61.86%) (false positives) | 46,607 (all successful interactions) |
| **Unsuccessful interactions** | 29,263 (42.25%) (false negatives) | 39,998 (57.75%) | 69,261 (all unsuccessful interactions) |

## 4.2 Implicit User Preferences

**Data**

To evaluate the predictive power of the implicit preferences we consider users who have at least 3 successful and 3 unsuccessful interactions during a one-month period (February 2010). This dataset was chosen so that we could test on the March dataset used in the study of the implicit preferences above. Here too, we restrict this subset to users who reside in Sydney. These two requirements are satisfied by 3,881 users, called *target users*. The training data consists of the interactions of the target users during February; 113,170 interactions in total, 30,215 positive and 72,995 negative. The test data consists of the interactions of the target users during Match; 95,777 interactions in total, 34,958 positive (37%, slightly less than the 40% in the study above) and 60,819 negative (63%, slightly more than the 60% in the study above). Each target user $U$ has a set of *interacted users* $U_{int}$, consisting of the users $U$ had interacted with.

**Method**

For each target user $U$ we create a classifier by training on $U$'s successful and unsuccessful interactions from February as described in Section 3.2. We then test the classifier on $U$'s March interactions. This separation ensures that we are not training and testing on the same interactions.

**Results**

Table 2 summarizes the classification performance of the NBTree classifier on the test data. It obtained an accuracy of 82.29%, considerably higher than the ZeroR baseline of 63.50% and the accuracy of the explicit preferences classifier. In comparison to the explicit preferences, the false positives drop from 61.86% to 30.14%, an important improvement in this domain since a recommendation that leads to rejection can be discouraging; the false negatives drop from 42.25% to 9.97%.

**Table 2.** Classification performance of NBTree on test set

| | **Classified as:** **Successful interactions** | **Unsuccessful interactions** | **Total** |
|---|---|---|---|
| **Successful interactions** | 24,060 (68.83%) | 10,538 (30.14%) (false positives) | 34,958 (all successful interactions) |
| **Unsuccessful interactions** | 6,064 (9.97%) (false negatives) | 54,755 (90.03%) | 60,819 (all unsuccessful interactions) |

In summary, the results show that the implicit preferences are a very good predictor of the success of user interactions, and significantly more accurate than the explicit preferences.

# 5 Using User Preferences in Recommender Systems

In this section, we propose an approach for using the implicit and explicit user preferences in a recommender system. More specifically, we proposed that they are used to rank the recommendation candidates in our hybrid content-collaborative recommender system [6]. We evaluate the performance of the two methods, compare them with a baseline and also with the currently used ranking method Support which does not use user preferences.

## 5.1 Hybrid Content-Collaborative Reciprocal Recommender

In [6] we described our hybrid content-collaborative reciprocal recommender for online dating. It uses information from the user profile and user interactions to recommend potential matches for a given user. The content-based part computes similarities between users based on their profiles. The collaborative filtering part uses the interactions of the set of similar users, i.e. who they like/dislike and are liked/disliked by, to produce the recommendation. The recommender is *reciprocal* as it considers the likes and dislikes of both sides of the recommendation and aims to match users so that the paring has a high chance of success.

The process of generating an ordered list of recommendations for a given user $U$ comprises of three key steps, see Fig. 2:

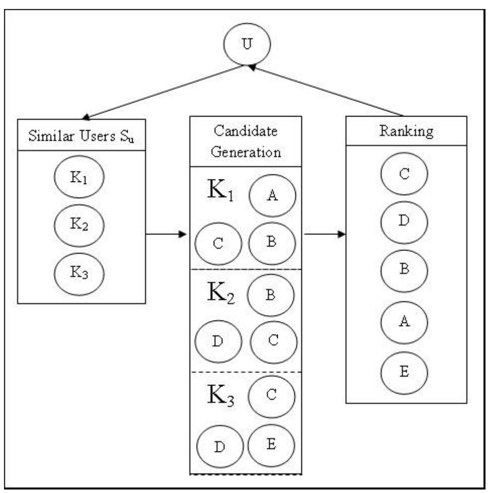


**Fig. 2.** Recommender process

**1. Similar User Generation Based on User Profiles**

This step produces a set of $K$ users who have the lowest possible distance to $U$. For example, in Fig. 1 the set of similar users $Su$ for user $U$ consists of $K_1$, $K_2$ and $K_3$. We use a modified version of the K-Nearest Neighbor algorithm, with seven attributes (*age, height, body type, education level, smoker, have children* and *marital status*) and a distance measure specifically developed for these attributes as described in [6].

**2. Candidate Generation Based on User Interactions**

This step produces a set of candidate users for recommending to *U*. For every user in *Su*, we compute the list of all users with whom they have reciprocal interest with, meaning that these people like *U* and are also liked by *U*. For example in Fig. 1 the recommendation candidate set for *U* is *{A, B, C, D, E}*.

**3. Candidate Ranking**

This step ranks the candidates to provide meaningful recommendations for *U*. In [6] we use an approach based on the user interactions, in particular the support of *Su* for the candidates. In this paper we propose two new ranking approaches based on the user preferences: Explicit and Implicit which are based on explicit and implicit user preferences, respectively. We describe the ranking methods in the next section.

## 5.2 Ranking Methods

### 5.2.1 Support

This ranking method is based on the interactions between the group of similar users *Su* and the group of candidates. Users are added to the candidate pool if they have responded positively to at least one *Su* user or have received a positive reply from at least one *Su* user. However, some candidates might have received an EOI from more then one *Su* user and responded to some positively and to others negatively. Thus, some candidates have more successful interactions with *Su* than others. The Support ranking method computes the support of *Su* for each candidate. The higher the score, the more reciprocally liked is *X* by *Su*. This ranking method is the method used in [6].

For each candidate *X* we calculate the number of times *X* has responded positively or has received a positive response from *Su*, see Table 3. We also calculate the number of times *X* has responded negatively or has received a negative response from *Su*. The support score for *X* is the number of positive minus the number of negative interactions. The higher the score for *X*, the more reciprocally liked is *X* by *Su*. The candidates are sorted in descending order based on their support score.

**Table 3.** Support ranking - example

| Candi-date | # Positive responses of candidate to *Su* | # Positive responses of *Su* to candidate | # Negative responses of candidate to *Su* | # Negative responses of candidate to *Su* | Support score |
|---|---|---|---|---|---|
| ***A*** | 10 | 1 | 4 | 2 | 5 |
| ***B*** | 4 | 2 | 4 | 1 | 1 |
| ***C*** | 5 | 1 | 1 | 1 | 4 |
| ***D*** | 2 | 0 | 6 | 1 | -5 |

### 5.2.2 Explicit

This ranking method is based on minimizing the number of non-matching attributes between the candidate profile and the explicit preferences of the target user; the lower the number of non-matches, the higher the candidate ranking. In addition to checking if the candidate satisfies the target user's explicit preferences it also checks the reverse: if the target user satisfies the candidate's explicit preferences. Thus, it minimises the number of reciprocal non-matches.

**Table 4.** Explicit ranking - example

| Candidate | # Matching attributes | # Non-matching attributes | Stage 1: non-match rank | Stage 2 = final ranking: match rank for ties |
|---|---|---|---|---|
| ***A*** | 2 | 0 | 1 | 2 |
| ***B*** | 2 | 2 | 2 | 4 |
| ***C*** | 4 | 2 | 2 | 3 |
| ***D*** | 4 | 0 | 1 | 1 |

We compare each candidate (i.e. its profile) with the explicit preferences of the target user and each target user with the explicit preferences of the candidate. We tally the number of matches and non-matches from both comparisons. The candidates are first sorted in ascending order based on the non-match score (stage 1 ranking). After that candidates with the same non-match score are sorted in descending order based on their match score (stage 2 and final ranking). An example is shown in Table 4.

### 5.2.3 Implicit

This ranking method uses the classifier generated for each target user *U*, based upon *U*'s previous interactions. Given a candidate, the classifier gives a probability for the two classes *Success* and Failure (successful and unsuccessful interaction between the candidate and target user, respectively). Candidates are then ranked in descending order based on the probability of class *Success*. Table 5 shows an example.

**Table 5.** Implicit ranking - example

| Candidate | Probability for Class *Success* |
|---|---|
| ***A*** | 0.95 |
| ***B*** | 0.74 |
| ***C*** | 0.36 |
| ***D*** | 0.45 |

### 5.2.4 Baseline

This ranking method assumes that all candidates have an equal chance of a successful pairing and that any one random selection will give the same chance of success as any other ranking approach. For each candidate pool the candidates are randomly shuffled before being presented to the target user.

## 5.3 Experimental Evaluation

### 5.3.1 Data

We used the same data as the data used to learn the implicit user preferences, see Section 4.2. As stated already, it consists of the profile attributes and user interactions of all users who had at least 3 successful and at least 3 unsuccessful interactions during February 2010 and reside in Sydney.

For each run of the experiment, the users who meet the two requirements listed above are considered as part of the test set. Information about a test user's interactions is never included when generating and ranking candidates for that user. This ensures a clean separation between testing and training data.

### 5.3.2 Evaluation Metrics

For a user *U* we define the following sets:

- Successful EOI sent by *U, successful_sent*: The set of users who *U* has sent an EOI where the user has responded positively.
- Unsuccessful EOI sent by *U, unsuccessful_sent*: The set of users who *U* has sent an EOI where the user has responded negatively.
- Successful EOI received by *U, successful_recv*: The set of users who have sent an EOI to *U* where *U* has responded positively.
- Unsuccessful EOI received by *U, unsuccessful_recv*: The set of users who have sent an EOI to *U* where *U* has responded negatively.
- All successful EOI for *U*: *successful=successful_sent+successful_recv*
- All unsuccessful EOI for*U: unsuccessful=unsuccessful_sent+unsuccessful_recv*

For each user in the testing set, a list of *N* ordered recommendations *N_recommendations* is generated. We define the successful and unsuccessful EOI in the set of *N* recommendations as:

- Successful EOI for *U* that appear in the set of *N* recommendations: *successful@N = successful* ∩ *N_recommendations.*
- Unsuccessful EOI for *U* that appear in the set of *N* recommendations: *unsuccessful@N = unsuccessful* ∩ *N_recommendations.*

Then, the *success rate* at N (i.e. given the *N* recommendations) is defined as:

$$success\ rate@N = \frac{\#successful@N}{\#successful@N + \#unsuccessful@N}$$

In other words, given a set of *N* ordered recommendations, the success rate at *N* is the number of correct recommendations over the number of interacted recommendations (correct or incorrect).

Each experiment has been run ten times; the reported success rate is the average over the ten runs.

## 5.4 Results and Discussion

Fig. 3 shows the success rate results for different number of recommendations *N* (from 10 to 200) and different number of minimum EOI sent by *U* (5, 10 and 20). Table 6 shows the number of users in the test set for the three different EOI_sent.

The main results can be summarised as follows:

- The three recommenders (using Support, Implicit and Explicit as ranking methods) outperform the baseline for all *N* and minimum number of EOI.
- The best ranking method is Support, followed by Implicit and Explicit. For a small number of recommendations (*N*=10-50), Implicit performs similarly to Support. This is encouraging since the sucess rate for a small number of recommendations is very important in practical applications. As *N* increase the difference between Support and Implicit increases.

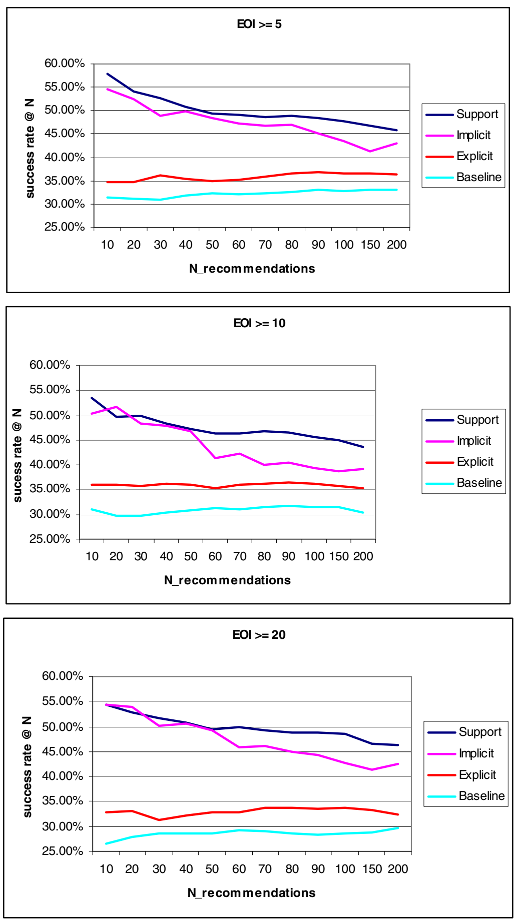


**Fig. 3.** Success rate for various *N* and minimum number of EOI sent by *U*

- Implicit significantly outperforms Explicit for all *N* and minimum number of EOI. For instance, when the top 10 recommendations are presented (*N*=10), the success rates are: Implicit=54.59%, Explicit=34.78% for EOI=5; Implicit= 50.45%, Explicit=36.05% for EOI=10; Implicit=54.31%, Explicit=32.95% for EOI=20, i.e. the difference between the two methods is 14.4-21.4%.
- As the number of recommendations *N* increases from 10 to 200, the success rate for Support and Implicit decreases with 8-12%. This means that the best recommendations are already at the top. Hence, these ranking methods are useful and effective. For Explicit as *N* increases the success rate doesn't change or even slightly increases in some cases which confirms that the rankig function is less effective, although still better than the baseline.
- A comparison of the three graphs show that as the number of EOI_sent increases from 5 to 20, the success rate trends are very similar.

**Table 6.** Number of users in the test set for the different number of EOI sent by *U*

| EOI_sent | Number of users |
|---|---|
| EOI_sent >=5 | 3,881 |
| EOI_sent >=10 | 3,055 |
| EOI_sent >=20 | 1,938 |

## 6 Conclusions

In this paper we have reported our study of user preferences in a large dataset from an Australian online dating website.

We first considered the explicit user preferences which consist of the stated characteristics of the preferred dating partner by the user. We showed that the explicit preferences are not a good predictor of the success of user interactions, achieving an accuracy of 49.43%. We found that 61.86% of all successful interactions are with people who do not match the user's explicit preferences and 42.25% of all unsuccessful interactions are with people who match the user's explicit preferences.

We then proposed a novel model of implicit preferences that is learned using a NBTree classifier from both successful and unsuccessful previous user interactions. We showed that it is a very good predictor of the success of user interactions, achieving an accuracy of 89.29%.

We also proposed an approch that uses the explicit and implicit preferences for ranking of candidates in an existing recommender system. The results show that both ranking methods, Explicit and Implicit, outperform the baseline and that Implicit is much more accurate than Explicit for all number of recommendations and minimum number of EOI we considered. For example, when the top 10 recommendations are presented and the minimum number of EOI sent is 5, the success rate of Implicit is 54.59%, the success rate of Explicit is 34.78% and the baseline success rate is 31.35%. In practical terms, the success rate for a small number of recommendations is the most important; for 10-50 recommendations Implicit performs similarly to the best ranking method Support that is not based on user preferences.

Users can benefit from a suitable presentation of their implicit preferences; they can compare the implicit and explicit preferences and adjust the explicit preferences accordingly. We plan to investigate this in our future work.

**Acknowledgments.** This work is supported by the Smart Services Cooperative Research Centre.

## References

1. Kim, Y.S., Mahidadia, A., Compton, P., Cai, X., Bain, M., Krzywicki, A., Wobcke, W.: People Recommendation Based on Aggregated Bidirectional Intentions in Social Network Site. In: Kang, B.-H., Richards, D. (eds.) PKAW 2010. LNCS, vol. 6232, pp. 247–260. Springer, Heidelberg (2010)
2. Cai, X., Bain, M., Krzywicki, A., Wobcke, W., Kim, Y.S., Compton, P., Mahidadia, A.: Collaborative Filtering for People to People Recommendation in Social Networks. In: Li, J. (ed.) AI 2010. LNCS, vol. 6464, pp. 476–485. Springer, Heidelberg (2010)
3. McFee, B., Lanckriet, G.R.G.: Metric Learning to Rank. In: 27th International Conference on Machine Learning, ICML (2010)
4. Diaz, F., Metzler, D., Amer-Yahia, S.: Relevance and Ranking in Online Dating Systems. In: 33rd Int. Conf. on Research and Development in Information Retrieval (SIGIR), pp. 66–73 (2010)
5. Pizzato, L., Rej, T., Chung, T., Koprinska, I., Yacef, K., Kay, J.: Learning User Preferences in Online Dating. In: European Conf. on Machine Learning and Priciples and Practice of Knowledge Discovery in Databases (ECML-PKDD), Preference Learning Workshop (2010)
6. Akehurst, J., Koprinska, I., Yacef, K., Pizzato, L., Kay, J., Rej, T.: A Hybrid Content-Collaborative Reciprocal Recommender for Online Dating. In: International Joint Conference on Artificial Intelligence, IJCAI ( in press, 2011)
7. Kohavi, R.: Scaling Up the Accuracy of Naive-Bayes Classifiers: a Decision-Tree Hybrid. In: 2nd International Conference on Knowledge Discovery in Databases, KDD (1996)
8. Cao, L.: In-depth Behavior Understanding and Use: the Behavior Informatics Approach. Information Science 180(17), 3067–3085 (2010)