THE UNIVERSITY OF SYDNEY

SCHOOL OF INFORMATION TECHNOLOGIES

LEARNING USER PREFERENCES IN ONLINE DATING

TECHNICAL REPORT 656

**LUIZ PIZZATO, THOMAS CHUNG, TOMASZ REJ, IRENA KOPRINSKA
KALINA YACEF AND JUDY KAY**

JULY, 2010

Learning User Preferences in Online Dating

Luiz Pizzato, Thomas Chung, Tomasz Rej,
Irena Koprinska, Kalina Yacef, and Judy Kay

School of Information Technologies
University of Sydney, NSW 2006, Australia
{forename.surname}@sydney.edu.au

Abstract. Online dating presents a rich source of information for preference learning. Due to the desire to find the right partner, users are willing to provide very specific details about themselves and about the people they are looking for. The user can describe his/her ideal partner by specifying values for a set of predefined attributes. This explicit preference model is quite rigid and may not reflect reality, as users' actions are often contrary to their stated preferences. In this respect learning implicit user preferences from the users' past contact history may be a more successful approach for building user preference models for use in recommender systems. In this study, we analyse the differences between the implicit and explicit preferences and how they can complement each other to form the basis for a recommender system for online dating.

1 Introduction

Online dating websites provide means for people to advertise themselves and to search through other people's advertised profiles. In order to find potential dating partners, users are willing to disclose information about who they are (their profile) and who they are searching for (their ideal partner profile).

Figure 1 shows an example of the user profile and ideal partner profile for Alice, a fictitious user. Alice's profile contains two types of information: constrained (attribute-value) and unconstrained (free text). The constrained part is selected from a list of options and represents information such as gender, age and location. The unconstrained part allows users to describe themselves and express their preferences (such as reading, music and movie tastes) in their own words.

The ideal partner profile also consists of constrained and unconstrained information. The constrained part of the ideal partner profile contains the same attributes as the constrained part of the user profile and can be used directly to find users who match the desired characteristics. In contrast, the unconstrained part is much harder to use to generate recommendations for the following reasons. First, it does not usually correlate with the textual profile of the user, e.g. users do not say "I want to date someone who likes romantic comedies and jazz". Second, it is typically vague, which represents a problem even for the most sophisticated natural language processing methods

Profile: Alice

Gender: Female	**Age**: 25	**Height**: 160 cm	**Weight**: 50 kg
Location: Sydney	**Smokes**: No	**Hair colour**: Blonde	**Eye colour**: Blue

About me: I'm a medical student interested in meeting a broader range of people - after this long, uni students, and associated lifestyle are losing appeal. I love to cook - going through an Italian phase at the moment, but I love eating out and experimenting with new restaurants too. I love any new experiences but admit to being a bit of a wimp when it comes to adventure involving heights and other adrenaline inducing factors. Silly dress up parties make me ridiculously happy, and I can't deal with people who take themselves too seriously.
Reading taste: Way too many text books to read at the moment. When I get the chance I like to read chic lit books and I also love a good crime novel.
Music taste: I love going to summer music festivals, live jazz or classical music.
Movie taste: Some of my favourite movies: Meet Joe Black, Transformers, The Proposal, Walk the Line, Crash, In Her Shoes, The Departed, Revolutionary Road. I also like TV shows like Grey's Anatomy, MasterChef, Friends, and Sex & the City.

Ideal Date

Gender: Male	**Age**: 25 – 30	**Height**: 170 – 190 cm	**Weight**: —
Location: Sydney	**Smokes**: No	**Hair colour**: —	**Eye colour**: —

Ideal date: Kind hearted, respectful, goal orientated, has character integrity, truthful, funny, able to see the lighter side of life, warm, considerate, someone who can openly communicate and articulate their feelings and wants. Also someone cultured, intelligent and worldly.

Fig. 1: Profile of the fictitious user Alice and information about her ideal date

Explicitly stating the characteristics of the ideal partner provides invaluable information about the users' likes and dislikes. Despite this, there are many cases where users do not provide detailed information about who they like. For instance, some online dating users are mostly reactive, meaning that they do not normally initiate communication with other users. These users do not provide explicit preferences so any indication of who they like or dislike is important. This motivates the use of implicit information extracted from their actions on the web site (contact history).

In this paper we present an approach for learning and representing implicit and explicit user preferences. We also study the differences between them and how they can be combined in a recommender system for online dating. For this study, we only use constrained attributes as they have clearer semantics.

Section 2 summarizes the relevant previous work on preference learning and the use of explicit and implicit preference profiles. Section 3 defines the two types of preferences we use, explicit and implicit, and describes our method for learning and representing them. Section 4 describes our preference-based recommender. The evaluation results are presented and discussed in Section 5. Finally, Section 6 presents the concluding remarks.

2 Related Work

Preference learning aims to find the preferences of a set of objects, by extrapolating known preferences of a similar, or possibly the same, set of objects [7]. Common tasks in preference learning include: classification, where a classification function maps each object to a set of predefined classes; and ranking [2, 17], where an object is classified into one of several ranked classes, or is ranked among other objects.

Preference learning methods generally fall under one of the two techniques: utility-based and relation-based. Utility-based methods (e.g. [12]) rank objects using a utility function which represents the degree of usefulness of the object. In contrast, relation-based methods (e.g. [15]) aim to find a function on pairs of objects which returns the more preferable object of the pair. While relation-based methods require fewer assumptions than utility-based methods, such as the existence of a transitive preference [15] or total ordering on the entire set, they are less convenient to utilize than a utility function when a ranking is desired.

Preference learning is an integral part of recommender systems, as recommendations are generated based on perceived preferences of the user. Two main paradigms for recommender systems are *Content Based* and *Collaborative Filtering*. Content based recommenders create user and item profiles based on their attributes (e.g. demographic information for users, price and quality for items etc), while collaborative filtering creates profiles based on links between users and items from previous interactions (e.g. users' previous rated/purchased items). Most recommender systems do not fall strictly into one category but use a combination of these techniques. An overview of preference learning in recommender systems is presented in [5].

Recommender systems require input from users to train on in order to capture users' preferences. Users may be explicitly asked to elucidate their preference or the recommender system could infer their preference implicitly by observing their actions. In the field of recommender systems, explicit feedback is often assumed to be superior to implicit feedback in terms of accuracy and hence predictive power [1], and is often the only reference to which implicit feedback is compared to. However, studies from other fields such as psychology [6] presents evidence otherwise.

The natural variability in the input provided by the user creates a "magic barrier" for the performance of recommenders has been suggested [8, 9]. Amatriain et al. [1] studied the reliability of user surveys and stability of users' judgements on users of the Netflix database and concluded that surveys are a reliable instrument and user opinions are stable. However, an earlier study by Hill et al. [9] with a smaller set of users in an online community reported a much lower value for the stability of user ratings. Cosley et at. [4] also reported a lower value of user ratings stability and demonstrated that users' ratings can be influenced by the user interface. It is not clear whether the accuracy and stability of user provided input is domain dependent. These results point to the need for a supplement or replacement to explicit feedback, such as implicit feedback.

Implicit feedback has been studied as an alternative to explicit feedback as input for recommender systems [14], motivated by the fact that users feel burdened when required to provide feedback to a recommender system and that collecting large amounts of training data requires no extra effort from the user [11, 13]. A survey of the types of implicit feedback in different domains can be found in [11].

Positive results regarding the use of implicit feedback have been reported for classifying URL mentions on Usenet [10], ranking relevance of query strings [18] and predicting interest in webpages [3].

Both explicit and implicit feedback have a role to play in generating useful recommendations. Explicit feedback is invaluable to make the first recommendations addressing the cold start problem. In addition, if a recommender system ignores explicit feedback, many users who have taken the trouble to specify such preference would be frustrated. Likewise, implicit preferences, learned by observing the behaviour of the user, would be expected to be taken into account for future recommendations. Hence, we need to gain a greater understanding of the relative performances of implicit and explicit preferences and investigate suitable ways of combining them. In this paper, we re-examine the perceived superiority of explicit feedback over implicit feedback in the domain of recommender systems. We build and evaluate a recommender based on explicit and implicit feedback for online dating.

3 User Preferences

In this section we describe the context of our study and the two different data sources for learning users' preferences.

3.1 Domain Overview

We have implemented our recommender system using data from a major online dating website.

When a user u signs up for an account on the website, he/she is asked to provide information about himself/herself using a set of predefined attributes such as age, height, occupation. This information forms the user profile. The user u may then contact other people by first filling in a search form describing the attributes of people he/she would like to contact, which returns a ranked list of users who match the description. The user u may then browse through this ranked list and decide whether to contact any of these users.

If u decides to contact another user v, he/she can choose a message from a list of predefined messages to send to v. The predefined messages are typically complements and indicate interest in further communication. The receiver v can choose a predefined response that can be positive or negative or decide not to respond at all. If v responds positively, then we call the message a *successful message*.

Anytime in this process but usually after a successful message, u may purchase a token from the website allowing him/her to send an unmediated message to v which is the only way for the two users to exchange contact details and develop further relationship.

3.2 User Profile

A user's profile consists of a list of attribute values. Most of the attributes are nominal, e.g. the *body type* attribute can take one of the following values: "slim", "athletic", "normal", "overweight" and "obese". There is also a small number of continuous attributes such as *age* and *height* that we transformed into nominal using binning.

3.3 Implicit Preferences

Implicit preferences of a user are the preferences learned from the actions of the user. Actions which indicate interest in another user include: viewing the user's profile, sending a message, replying positively to a message received or purchasing a token in order to send an unmediated message. To learn a user's implicit preferences, we have chosen two actions: sending a message and replying positively to a message. These two actions are the strongest indicators for an interest in another user as viewing a profile can be done without specific interest in the user and purchasing a token is rarely done without first sending a message or replying positively to a message.

Consider a user u. Denote the set M_u to be the message history of u:

$$M_u = \{v : u \text{ messaged } v \text{ or } v \text{ messaged } u \text{ and } u \text{ responded positively}\}$$

For each attribute, we find the distribution of attribute values over all users in M_u. The collection of these distributions over all attributes is defined to be the implicit preferences of u. Figure 2 shows an example of implicit user preferences for four attributes. The implicit preferences effectively summarise u's message history and are learned without solicitation from the user.

3.4 Explicit Preferences

In contrast to implicit preferences, explicit preferences are acquired by explicitly asking the user to tell the system what he/she likes by completing the ideal partner profile. Thus, the explicit preferences consist of the ideal partner profile attribute values. Instead of distribution of values we use binary representation as it is not clear how much an attribute value appeals to a user as discussed in the next section.

3.5 Comparison of Implicit and Explicit Preferences

Both implicit and explicit preferences consist of attribute values which appeal to u. When dealing with explicit preferences we do not have information about how important an attribute value is to u. In the online dating site we are working with, the user can specify that he/she likes an attribute value but not how much. Implicit preferences, on the other hand, give some indication of the user's preference for certain attribute values. Consider the case when u has messaged

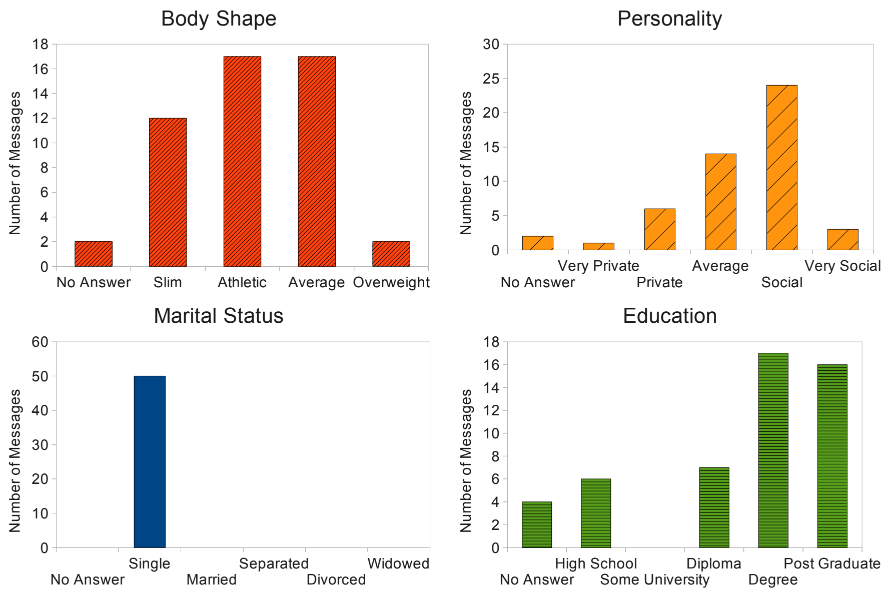


Fig. 2: Implicit user preferences

three slim and one overweight users; we can use this information to derive a preference ranking function for implicit but not explicit preferences.

As the message history for u gets larger, it makes more sense to rely on the implicit preferences as the users' explicit preferences may be incomplete or unreliable. We have found that in our dataset 10% of users did not define an age range in their explicit profile. Furthermore, we have found that in general people's explicit preferences are not as specific as their implicit preferences. In many cases they will specify a wide age range, such as 18-30, but only message people within a subset of that range, such as 24-27. Presenting the implicit preferences to a user and contrasting them with his/her explicit preferences, could allow the user to better understand his/her true preferences in terms of attributes and improve his search criteria for future searches.

4 Preference-Based Recommender System

To generate recommendation (a ranked list of users) for a given user u, our system follows a four-step process. First, it creates u's implicit and explicit preference models. Second, it filters out users who do not match the preference models. Third, it generates the recommendation list based on the selected approach: using only implicit or explicit preferences or combining both of them. Fourth, it ranks the candidates in the recommendation list using ranking criteria and presents the top n recommendations to the user. Step 1 has already been discussed in the previous section; here we discuss the remaining steps.

4.1 Filtering Users

When considering potential matches for u, we can automatically exclude a large number of users by filtering out all users who do not match u's preferences. For example, if u is heterosexual we do not need to consider users of the same gender as potential candidates. The filtering can be done based on the implicit or explicit preferences.

Consider an attribute A_k, with values a_1, a_2 and a_3. Assume that only a_1 and a_2 are included in u's preferences. If some user v's profile includes a_3 as the value for attribute A_k, then v will be filtered out of u's recommendation list. More generally v will be filtered out if, within all N attributes of a user profile, there exists some attribute A_k, $k \in \{1, \ldots, N\}$ such that a_l is the value of the attribute in v's profile, but a_l is not included in u's preferences. All users who pass the filtering stage are said to be in u's recommendation list. The filtering stage filters out on average 95% of candidates in our experiments.

4.2 Generating Recommendation Lists Using Implicit, Explicit, Intersect and Switch Methods

The filtering step produces one or two recommendation lists: a list based on the implicit preferences and a list based on the explicit preferences. We use these lists separately (Implicit and Explicit methods, respectively) and also combine them using two methods (Intersect and Switch methods). The Intersect method generates the intersection of the two recommendation lists. More specifically, the pair (u, v) will be in the intersection if (u, v) is in the implicit recommendation list and (u, v) is in the explicit recommendation list. The Switch method uses the implicit preferences for users who have sent more messages than a threshold m and the explicit preferences for the remaining users.

4.3 Ranking

It is likely that users will receive several hundred candidates in their recommendation list. It is unreasonable to expect users to view all these candidates so some sort of ordering is necessary, where the best predicted matches appear at the top of the list. To rank the candidates we have chosen to use our *Reciprocal Compatibility Score* [16].

The reciprocal compatibility score $recip\_compat(u, v)$ for users u and v is the harmonic mean of the compatibility scores of these users, $compat(u, v)$ and $compat(v, u)$:

$$recip\_compat(u, v) = \frac{2}{compat(u, v)^{-1} + compat(v, u)^{-1}}$$

The compatibility score gives an estimate of how compatible two users are to each other. Assume that each user has N attributes $\{A_1, \ldots, A_N\}$ and each attribute A_i takes k_i values from the set $\{a_{i1}, a_{i2}, \ldots, a_{ik_i}\}$. We define the frequency in occurrence of an attribute value in u's implicit preference as $f_{u,i,j}$, for

Profile	Alice	Bob
Gender	Female	Male
Age	23	26
Body	Slim	Athletic

Preferences	Alice	Bob
Gender	(Male, 20)	(Female, 9) (Male, 1)
Age	(25-29, 5) (30-34, 10) (35-40, 5)	(20-24, 3) (25-29, 6) (30-34, 1)
Bodytype	(Athletic, 18) (Average, 2)	(Athletic, 5) (Average, 4) (Slim, 1)

Fig. 3: Sample profile and preferences of two users: Alice and Bob

some attribute A_i and some attribute value a_{ij}. We define the profile function $P(u, i, j)$, for some user u, some attribute A_i and some attribute value a_{ij} as:

$$P(u, i, j) = \begin{cases} 1 & \text{if } a_{ij} \text{ in } u\text{'s profile} \\ 0 & \text{otherwise} \end{cases}$$

The compatibility score, *compat*, for a pair of users is defined as:

$$compat(u, v) = \sum_{i=1}^{n} \sum_{j=1}^{k_i} \frac{f_{u,i,j}}{k_i} \times P(v, i, j)$$

As an example consider the users Alice and Bob in Figure 3. For illustrative purposes their profiles are represented by only three attributes: gender, age and bodytype. The preferences are given in terms of the attribute values and frequencies of these attribute values. We calculate the compatibility score of Bob to Alice as follows:

$$compat(Bob, Alice) = \frac{f_{Bob,Gender,Female}}{k_{Gender}} + \frac{f_{Bob,Age,(20-24)}}{k_{Age}} + \frac{f_{Bob,Bodytype,Slim}}{k_{Bodytype}}$$

$$compat(Bob, Alice) = \frac{9}{10} + \frac{3}{10} + \frac{1}{10} = 0.43$$

Note that the compatibility score only works with implicit preferences. As mentioned already when dealing with explicit preferences it is unclear if a user prefers one attribute value over another, assuming both attribute values appear in the user's preferences. However if u has both implicit and explicit preferences, as in our case, then it is possible to filter u's recommendation list using his explicit preferences and order his recommendation list using the compatibility score.

In this section we have shown how we can utilise explicit and implicit preferences to build a recommender system for online dating. In the following section we evaluate the recommender with the aim of answering the following question: When should implicit and explicit preferences be favoured over each other?

5 Evaluation

5.1 Evaluation Setup

Table 1 summarizes the training and testing data that we used. The training data consists of the user profile and interactions of all users who had sent a message or replied positively to a message within a one month period (24 March-24 April 2009, called training period). To learn the implicit preferences of these users we used the messages they sent during the training period. The explicit preferences of these users consisted of their stated ideal partner profile at the end of the training period. The testing data consists of all messages sent between users who were active during the training period and were sent up to one year before or after the training period. In order to compare implicit to explicit preferences, in our evaluation we have only included users who have both implicit and explicit preferences.

It is well documented that location is an important consideration for a user seeking a date. However, the location field of the users' ideal partner profile was mainly unspecified, we felt that for fair comparison we should ignore this attribute. Consequently, our training data only takes into consideration messages between users who live in Sydney, and our recommender is able to recommend Sydney users to other Sydney users.

The baseline is defined by the chance of a user to have a successful interaction with another user. A successful message occurs when a user sends a message to another user who replies positively. On the other hand a unsuccessful message occurs when a negative reply or no reply is sent. The baseline value for success rate is 14.1%, with nearly 1.5 million messages between users and 210 thousand messages replied to positively.

Our evaluation is based on the following metrics:

- **Precision of Success**: the percentage of successful messages among all recommendations given. This is the number of people recommended to User u that u actually messaged *successfully*, summed over all users, divided by the total number of recommendations.
- **Recall of Success**: the percentage of all known successful messages that were found among the recommendations given.
- **Success Rate**: the ratio of the number of successful recommendations to the number of recommendations for which we have an indication if they were positive or negative. In other words, it shows the percentage of successful messages among those messages that were correctly predicted.

Table 1: Training and Testing Data

	Training data	**Testing data**
Period	24/03 - 24/04/2009	24/04/08 - 01/03/2010 (excluding training period)
Users	21,430	21,430
Messages	362,032	1,430,931
Successful Messages	60,718	231,809

5.2 Results

Success Rate. Table 2 summarizes the success rate results for the Explicit and Implicit methods and compares them with the baseline (see Sec. 5.1). The results show that both methods outperformed the baseline of 14.1% and Implicit performed slightly better than Explicit. This suggests that using people's contact history is more reliable than using their stated preferences.

Table 2: Success rate for Explicit and Implicit methods

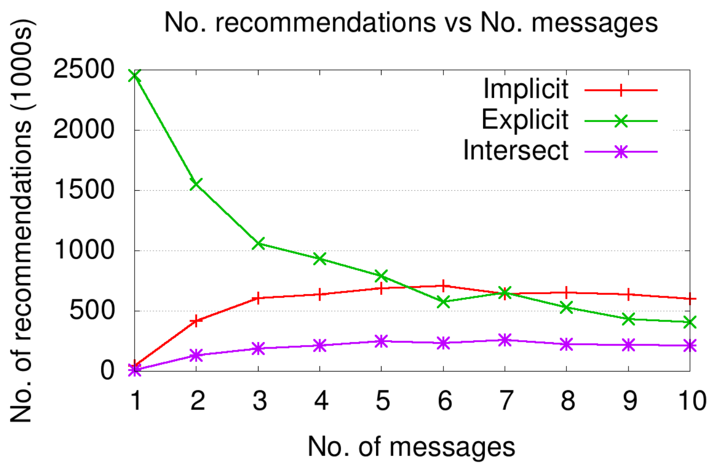
	Explicit	**Implicit**
Number of successful messages	54,507	95,452
Number of total messages	334,324	562,470
Success rate	16.3%	17.0%

Effect of the Number of Messages Sent. Figure 4 shows the number of recommendations generated and the precision of success for each filtering method, for different number of training messages. The number of training messages is the number of messages sent by a user within the training period.

For the number of recommendations generated, the average number of recommendations per user is roughly constant for the Explicit method, while for the Implicit method it is much lower for fewer messages as it is harder to find user preferences with fewer messages to train on. While the number of recommendations generated from the two methods have a decreasing trend as the number of messages becomes larger, their intersection remains roughly the same size,

Table 3: Number of successful messages predicted for different numbers of training messages and filtering methods

No. messages	1	2	3	4	5	6	7	8	9	10
Implicit	80	641	992	1074	1465	1591	1550	1676	1844	1537
Explicit	2309	1794	1326	1150	1385	1056	1330	1294	1172	977
Intersect	19	249	322	366	623	571	679	824	788	637



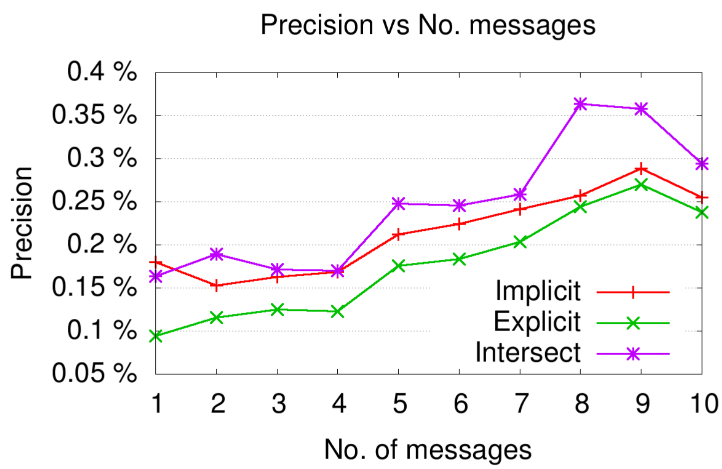


Fig. 4: Number of recommendations generated and the precision of success for different number of training messages

suggesting that the Implicit and Explicit methods converge for large number of messages sent. Table 3 shows the number of successful messages predicted for different numbers of training messages, and from it we can see the same trend for the Implicit and Explicit methods to converge.

In terms of precision of success, the Intersect method is the best, followed by the Implicit and then the Explicit methods. While we expect the precision of success for Explicit not to depend on the number of training messages, we do observe an increasing trend as the number of messages increases. This trend is possibly random fluctuation as the numbers involved are small. This however does not affect our conclusion on the relative performances of the different methods which remains the same for all data points. From Table 3 together with Figure 4 we can confirm the increased precision of the Intersect method. The Implicit and Explicit methods, although having a lower precision, are both valuable as they recommend a larger number of successful messages.

Effect of the Number of Recommendations. We compare the performance of the four recommendation approaches for different values of N, the number of the top recommendations presented to the user.

Figure 5 shows the precision and recall of correctly predicted successful messages for the for top-N recommendations for the four recommendation approaches. For precision, Intersect is the best approach for $N \leq 3$, followed by Implicit for $N \geq 4$. For recall, Implicit is the best approach for all N.

On both performance measures, the best performing approach is Implicit, very closely followed by Switch. This points to the value of implicit preferences for use in this class of recommender systems. Explicit is the worst performing approach, although its performance is closer to the other approaches for small N.

6 Conclusions

We presented approaches for learning explicit and implicit user preferences for a recommender system for online dating. Our results showed that the implicit preference model (based on user's activity) outperformed the explicit preference model (based on stating the characteristics of the ideal partner). Thus, for domains such as online dating, implicit preferences are a better representation of the actual user's preferences than explicit preferences. Combining implicit and explicit preferences is also a promising approach, with the Intersect method yielding a higher precision than either Implicit or Explicit separately.

The results of this study are consistent with our previous work [16] where we showed that explicit and implicit preferences only partially overlap. Implicit preferences are not simply a refinement of explicit preferences; users message other users who do not match the profile of their ideal partner. These differences between the implicit and explicit preferences can be explained with the difficulty in creating an accurate ideal partner profile. In the domain of online dating, people are *aware* that there are things they cannot accurately specify, such as whether it actually matters to them if a potential partner likes certain movie genres or has had a particular level of education. At the same time people may also be *unaware* of the importance of attributes of their ideal partner, for example, thinking they want a tall partner when it is not particularly important. The problem of inaccurate explicit user preferences is not confined to online dating; it is also a problem for all domains in which users do not know precisely what they want or are unable to accurately specify their preferences (e.g. specifying a query for web search).

Online dating is representative of an important class of systems which match people to people, such as matching mentors with mentees and matching job

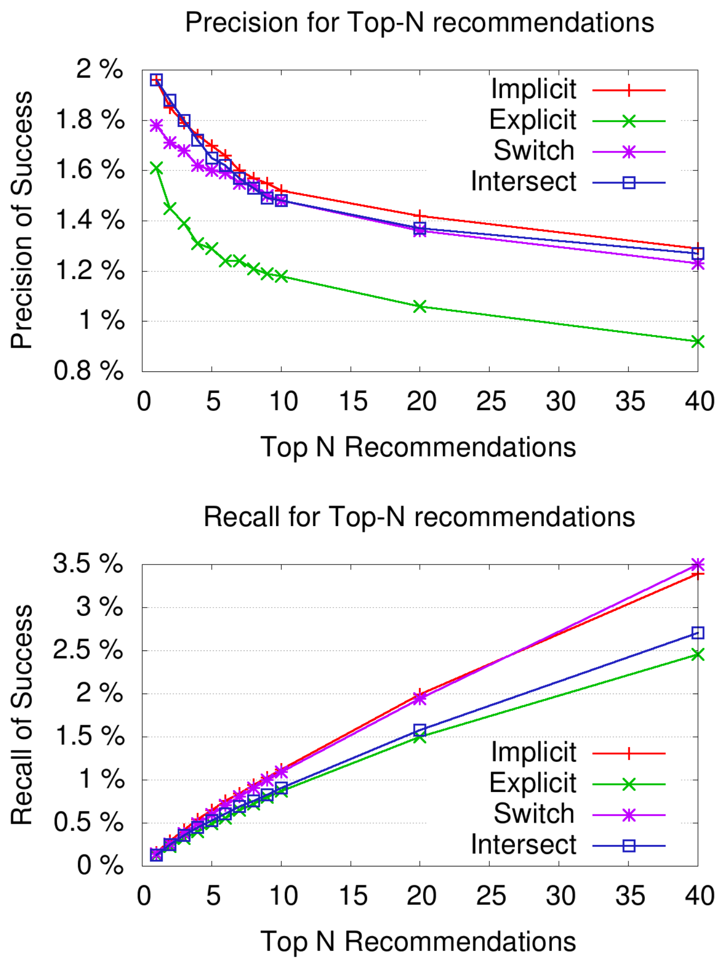


Fig. 5: Precision and recall of successful messages for top-N recommendations

applicants with employers, so our work is of broader relevance than just online dating. It would be interesting to investigate to what extent our conclusions about the importance of implicit preferences apply to these domains.

Acknowledgements

This research was funded by the Smart Services Co-operative Research Centre.

References

1. X. Amatriain, J. M. Pujol, and N. Oliver. I like it... i like it not: Evaluating user ratings noise in recommender systems. In *UMAP '09: Proceedings of the 17th International Conference on User Modeling, Adaptation, and Personalization*, pages 247–258, Berlin, Heidelberg, 2009. Springer-Verlag.
2. R. Caruana, S. Baluja, and T. Mitchell. Using the future to 'sort out' the present: Rankprop and multitask learning for medical risk evaluation. In *NIPS '95: Proceedings of Neural Information Processing Systems conference*, pages 959–965, 1995.
3. M. Claypool, P. Le, M. Wased, and D. Brown. Implicit interest indicators. In *IUI '01*, pages 33–40, New York, 2001. ACM.
4. D. Cosley, S. K. Lam, I. Albert, J. A. Konstan, and J. Riedl. Is seeing believing?: how recommender system interfaces affect users' opinions. In *CHI '03: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 585–592, New York, NY, USA, 2003. ACM.
5. M. de Gemmis, L. Iaquinta, P. Lops, C. Musto, F. Narducci, and G. Semeraro. Preference learning in recommender systems. In *Preference Learning (PL-09) ECML/PKDD-09 Workshop*, 2009.
6. A. Fiore, L. Shaw Taylor, X. Zhong, G. Mendelsohn, and C. Cheshire. Whom we (say we) want: Stated and actual preferences in online dating, 2010.
7. J. Fürnkranz and E. Hüllermeier. Preference learning. In C. Sammut and G. I. Webb, editors, *Encyclopedia of Machine Learning*. Springer-Verlag, To appear.
8. J. L. Herlocker, J. A. Konstan, L. G. Terveen, and J. T. Riedl. Evaluating collaborative filtering recommender systems. *ACM Trans. Inf. Syst.*, 22(1):5–53, 2004.
9. W. Hill, L. Stead, M. Rosenstein, and G. Furnas. Recommending and evaluating choices in a virtual community of use. In *CHI '95: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 194–201, New York, NY, USA, 1995. ACM Press/Addison-Wesley Publishing Co.
10. W. Hill and L. Terveen. Using frequency-of-mention in public conversations for social filtering. In *CSCW '96: Proceedings of the 1996 ACM conference on Computer supported cooperative work*, pages 106–112, New York, NY, USA, 1996. ACM.
11. D. Kelly and J. Teevan. Implicit feedback for inferring user preference: a bibliography. *SIGIR Forum*, 37(2):18–28, 2003.
12. T. Kliegr. Uta - nm: Explaining stated preferences with additive non-monotonic utility functions. In *Preference Learning (PL-09) ECML/PKDD-09 Workshop*, 2009.
13. D. M. Nichols. Implicit rating and filtering. In *Proceedings of the Fifth DELOS Workshop on Filtering and Collaborative Filtering*, pages 31–36, 1997.
14. D. W. Oard and J. Kim. Implicit feedback for recommender systems. In *AAAI Workshop on Recommender Systems*, pages 81–83, 1998.
15. T. Pahikkala, W. Waegeman, E. Tsivtsivadze, T. Salakoski, and B. De Baets. From ranking to intransitive preference learning: rock-paper-scissors and beyond. In *Preference Learning (PL-09) ECML/PKDD-09 Workshop*, 2009.
16. L. Pizzato, T. Rej, T. Chung, I. Koprinska, and J. Kay. Recon: A reciprocal recommender for online dating. In *RecSys*, to appear.
17. E. Tsivtsivadze, B. Cseke, and T. Heskes. Kernel principal component ranking: Robust ranking on noisy data. In *Preference Learning (PL-09) ECML/PKDD-09 Workshop*, 2009.
18. R. White, I. Ruthven, and J. M. Jose. The use of implicit evidence for relevance feedback in web retrieval. In *Proceedings of the 24th BCS-IRSG European Colloquium on IR Research*, pages 93–109, London, UK, 2002. Springer-Verlag.

ISBN 978-1-74210-193-4

School of Information Technologies
Faculty of Engineering & Information Technologies
Level 2, SIT Building, J12
The University of Sydney
NSW 2006 Australia

T +61 2 9351 3423
F +61 2 9351 3838
E sit.information@sydney.edu.au
sydney.edu.au/it

ABN 15 211 513 464
CRICOS 00026A