Finding someone you will like and who won't reject you

Luiz Pizzato, Tomek Rej, Kalina Yacef, Irena Koprinska, and Judy Kay

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

Abstract. This paper explores ways to address the problem of the high cost problem of poor recommendations in reciprocal recommender systems. These systems recommend one person to another and require that both people like each other for the recommendation to be successful. A notable example, and the focus of our experiments is online dating. In such domains, poor recommendations should be avoided as they cause users to suffer repeated rejection and abandon the site. This paper describes our experiments to create a recommender based on two classes of models: one to predict who each user will like; the other to predict who each user will dislike. We then combine these models to generate recommendations for the user. This work is novel in exploring modelling both people's likes and dislikes and how to combine these to support a reciprocal recommendation, which is important for many domains, including online dating, employment, mentor-mentee matching and help-helper matching. Using a negative and a positive preference model in a combined manner, we improved the success rate of reciprocal recommendations by 18% while, at the same time, reducing the failure rate by 36% for the top-1 recommendations in comparison to using the positive model of preference alone.

1 Introduction

Modelling what users *like* has allowed recommender websites to provide personalised recommendations of products they might want to purchase. There has been relatively little work that explicitly modelled negative preferences, independently of the positive model. This issue is somewhat subtle, because recommenders make use of both positive and negative ratings to build their preference models. These models are successful if they are effective in providing recommendations that have a high proportion of good recommendations. It may not matter if there is a small proportion of poor recommendations within this set.

However, for recommenders that focus on matching people to people, the cost of a poor recommendation can be quite high. This type of recommender is called *reciprocal recommender* [9] because it involves establishing reciprocal relationships between people in domains such as in online dating sites, employment websites (which aim to match employees and employers), mentor-mentee matching and matching helper and helpees. So, for example, consider a scenario where a user, *Bob*, is recommended to another user, *Alice*; this recommendation is only successful if both *Alice and Bob* reciprocally agree that the recommendation is good. Importantly, the interaction is staged. At the first stage, it is like other recommenders in the fact that *Alice* is presented with a set of recommendations and she can simply ignore the one for *Bob* if she does not like that recommendation (*Bob* may never know that he was recommended). However, it can be highly costly to the system if *Alice* initiates a contact with *Bob* and he then rejects her. If the same situation happens repeatedly, it may cause *Alice* to feel the anguish of repeated rejection.

Because the cost of poor reciprocal recommendations can be high, it seems likely that it will be valuable to build a model of negative preferences for such domains. Some key attributes of reciprocal recommenders make this seem rather important, compared to conventional recommenders. Notably, it is important to avoid overloading any individual. This is partly because each person only needs a small number of recommendations of people they should consider more seriously, for example, moving to establishing contact. It is also important from the recommendee perspective as it increases the risk of them being ignored or rejected by a popular person. Another key property of reciprocal recommenders is that they must involve every user in recommendations: every user must be given recommendations, and the system should recommend every user to others, no matter how unpopular they might be. This might mean that, compared with conventional recommenders, the reciprocal recommender may need to find recommendations that may not be a particularly good match to the user's preference model; in this case, the explicit modelling of negative preferences, may help avoid making recommendations that are more likely to incur the risk of rejection.

In this paper, we explore the impact of building and combining positive and negative preference models for online dating. We need to introduce some terminology for this context. The first stage of the recommendation process involves presenting a user (such as *Alice* in our scenario) with a set of recommendations. If this is successful, and the user likes one or more of the recommendations, the user can send an expression of interest (EOI) to the people they like. The EOI for our system is one of a set of short, pre-defined messages. The second stage involves the person who receives an EOI (*Bob* in our scenario), who can respond to an EOI with one of a small set of pre-defined messages. The response of an EOI can be either *positive* or *negative* indicating whether the recipient *likes* or *dislikes* the person who sent the EOI. At this second stage, a positive response indicates the recommendation process is proceeding successfully. The third key step is that one of the users can purchase a token allowing both users to exchange unconstrained messages that might contain contact details. After this stage users can meet face to face, or simply communicate outside the online dating website using standard electronic means.

The remainder of this paper is structured as follows. Section 2 reviews the literature in reciprocal recommenders for online dating and the use of negative indications of preference. Section 3 describes the main characteristics of the content-based reciprocal recommender used in this study, including the way we have incorporated negative indications of preferences. Section 4 presents the evaluation setup and the evaluation results, including how much negative preferences influenced the results, and especially its effect on preventing rejection. Finally, Section 5 discusses the results and their implications for other reciprocal recommenders.

2 Literature Review

Although online dating had not received much attention in recommender systems research in the past, the last year has seen several papers on the subject. These include our work [9] which introduced the notion or reciprocal recommenders, identified their distinctive characteristics and created RECON, a recommender we used to explore the effect of taking reciprocity into account. There was also the quite independent work of Diaz et al. [5] as well as McFee and Lanckriet [8], which focussed on finding a list of users whose chance of a positive interaction with another user is higher for those users near the top of the list than for those users near the bottom (i.e. a ranking problem). We will return to these in more detail after reviewing key work in modelling negative preferences.

There has been a range of research into use of negative preferences. For instance, the Adaptive Radio [4] is one example of work that explored the value of explicitly modelling negative preferences for group recommendation. The work of Kim et al. [7] creates a people recommendation system for a social network website where users can reply positively or negatively to other users. Kim et al. create groups of users based on common attributes and by comparing the communication between these groups, they find rules that can be applied to generate recommendations. They highlight that it is important to consider both sides of the recommendation process. They have shown that for the reciprocal domain of social networks, recommendation can improve by considering the preferences of the object of the recommendation. However, they did not report any particular use of the negative interactions, nor did they attempt to minimise negative responses.

Similarly, the collaborative filtering technique named SocialCollab [3] was also in the context of a social network. But they make no mention of negative interactions. Their algorithm combines two network of users: (1) users with similar "taste", and (2) users with similar "attractiveness". Similar taste is defined in terms of the users who send messages to the same group of users, while similar attractiveness relates to those users who receive messages from the same group of users. By combining these two strategies, they report improved performance, indicating the importance of reciprocity in a people to people recommender.

The work of Akehurst et al. [1] makes use of positive and negative responses to an EOI. Akehurst found that similar users, in terms of the attributes of the users, like and dislike similar groups of users. Using this information, Akehurst deals with the cold start problem by finding a set of users who were liked by a group of similar users, and used a ranking strategy that accounts for the number of positive and negative responses given by these users. This strategy means that the system aims to recommend people who send more positive replies and few negative replies to the set of similar users. This work used a hybrid approach, that combined collaborative filtering to generate recommendations, and content-based recommender algorithms to compute the list of similar users.

Brozovsky and Petricek [2] applied collaborative filtering in an online dating system. They used variations of user-user and item-item collaborative filtering algorithms and several different benchmarks using a dataset to predict the ratings that users would give for each other's appearance in an online dating service. The use of positive and negative indications are implicit in the ratings given by the user.

Taking an information retrieval and machine learning approach to finding good matches in an online dating scenario, McFee and Lanckriet [8] used structural support vector machine (SVM) to learn distance metrics optimised for different ranking measures. Because structural SVMs require positive and negative examples, McFee and Lanckriet used positive interactions as positive indications and then treated all other interactions (not necessarily known to be negative) as negative examples. Although McFee and Lanckriet reported better results than the baseline, the difference in the results is small.

Diaz et al. [5] focus on learning a reciprocal ranking function that maximises the chance of a positive interaction between online dating users. They describe the reciprocal aspects in the research as two-sided relevance. They used structured and unstructured profile features, including the information about the user's explicit preferences (the user "query") and the positive and negative interactions between users.

In this paper, we go beyond the class of the work discussed above by exploring the modelling of negative preferences, as well as positive preferences. We use these models to generate recommendations that we are most confident the user *likes* balanced by the need to present recommendations for people who are least likely to *dislike* the user.

3 RECON

As this work builds from the earlier RECON [9], we now describe it. RECON is a content-based reciprocal recommender system for online dating. It uses positive user interactions to build the model used to generate recommendations. RECON is a reciprocal recommender, meaning that it considers the preference models of both sides before a match is suggested to the user.

Consider the set of users with whom a user A has positively interacted. This is any user to whom A has sent an EOI or, the user has sent an EOI to A, who then replied positively. We extract the values of all profile features of these users. We store these values as a collection of counts and build a model of positive preferences M_A^+, which is then used to calculate the positive compatibility $C^+(A, B)$ of a user A with any user B. This positive compatibility is calculated by checking how many times each attribute value of B's profile occurred in M_A^+. These values are normalised by the number of users used to build the preference model and by the number of attributes in the user profile.

For instance, *Alice* has sent EOIs to 10 men with the following characteristics: 7 singles, 2 divorced, 1 separated; 5 smokers, 5 non smokers. The positive compatibility between *Alice* and a user *Bob* who is *single* and a *non smoker* is:

$$C^+(Alice, Bob) = \frac{7+5}{10 \times 2} = 0.6$$

The reciprocal part of the recommendation is created by finding the top-N highest reciprocal compatibility scores between all A, B pairs of users. The reciprocal positive compatibility scores is the harmonic mean between $C^+(A, B)$ and $C^+(B, A)$. Essentially, RECON learns who to recommend to a user *Alice*, by learning the people whom *Alice* is likely to like and, of these people, selecting those most likely to like her. It does this by making use only of the positive actions of *Alice* (people she has sent an EOI or replied positively to) and the positive actions of other users.

3.1 RECON using negative preferences

The same approach can be used create a model based on the negative interactions between the users (i.e. indications that someone does not like someone else). These negative models can be used to generate recommendations that are less likely to be disliked by the users.

Similarly to the positive preference model, given the set of negative user interactions of a user A, we define a negative preference model M_A^-. This negative preference model is used to calculate the negative compatibility $C^-(A, B)$ of a user A with any user B. This is essentially a model that measures the similarity between a user and the people whom *Alice* has negatively replied to their EOI.

Given a positive and negative compatibility, we can calculate the combined compatibility of a user A with a user B using A's positive and negative models of preference by subtracting the negative compatibility score from the positive compatibility score, with a normalisation step to obtain a compatibility score between 0 and 1. The formula is as follows:

$$C^{\pm}(A, B) = \frac{1 + C^+(A, B) - C^-(A, B)}{2} \quad (1)$$

In this way, it is possible to measure how much a user *Bob* matches the positive compatibility of a user *Alice* (i.e. how strongly the model predicts that *Alice* will like *Bob*) and how much *Bob* matches the negative compatibility of *Alice* (i.e. how strongly the model predicts that *Alice* will dislike *Bob*).

By combining both scores, we define a combined compatibility score that will give high scores for matches that are similar to the positive preference model and different from the negative preference model. Combined compatibility scores close to 0.5 are likely to be users who match the positive and the negative

Table 1. Example of recommendations and compatibility scores for a user x (ranking of recommendations for user x according to each compatibility score is shown in brackets)

$y =$	$C^+(x,y)$	$C^-(x,y)$	$C^\pm(x,y)$	$C^+(y,x)$	$C^-(y,x)$	$C^\pm(y,x)$	$C^+_{rec}(x,y)$	$C^\pm_{rec}(x,y)$
j	0.80 (1)	0.30 (2)	0.75 (1)	0.40	0.80	0.30	0.53 (2)	0.43 (4)
p	0.75 (2)	0.75 (5)	0.50 (3)	0.80	0.30	0.75	0.77 (1)	0.60 (1)
z	0.55 (3)	0.50 (3)	0.53 (2)	0.30	0.70	0.30	0.39 (4)	0.38 (5)
w	0.30 (4)	0.70 (4)	0.30 (5)	0.90	0.30	0.80	0.45 (3)	0.44 (3)
k	0.20 (5)	0.20 (1)	0.50 (3)	0.20	0.30	0.45	0.20 (5)	0.47 (2)

preference model equally. By contrast, combined scores close to zero indicate users who highly match someone's negative model and are a poor match to that person's positive model. Table 1 shows examples of these scores.

Similar to RECON, reciprocal recommendation can be created as the harmonic mean of the combined compatibility scores such that:

$$C^{\pm}_{rec}(A,B) = \frac{2}{\frac{1}{C^{\pm}(A,B)} + \frac{1}{C^{\pm}(B,A)}} \tag{2}$$

between all pairs of users A and B. We use harmonic mean because it is desirable to favour low compatibility scores over high scores when two users have distinctly different levels of compatibility. For instance, if *Bob* likes *Alice* a lot, and *Alice* does not like *Bob* at all, there is a very little chance that this reciprocal relationship will be successful; therefore, we want to have a reciprocal compatibility score more similar to *Alice*'s score than to *Bob*'s score.

In Table 1, we demonstrate how the values of the different compatibility scores relate to each other. For example, user p, who has a high positive compatibility score with user x (ranked second using $C^+(x,y)$) and a high negative score (ranked last[1] using $C^-(x,y)$), only ranks third in a combined score using $C^\pm(x,y)$. The same third position is occupied by k with a low $C^+(x,y)$ and a low $C^-(x,y)$. Also in the example of Table 1, we can observe that user j has the highest combined compatibility score with user x (highest $C^\pm(x,y)$); however because x's combined compatibility score with j (i.e. $C^\pm(j,x)$) is low, j has a low reciprocal compatibility score ($C^\pm_{rec}(x,y)$) in comparison to the other users in this example.

4 Evaluation

We have conducted our research in the context of one of the largest online dating websites in Australia. For the experiments described in this paper, we used one month of interactions (EOIs sent and their replies) between users to train our models, and the subsequent month to evaluate them. All experiments were clearly divided into training and testing data. For the purpose of evaluating the impact of negative preference model, we selected users who had both positive

[1] Notice that the negative ranking is from the lowest value to the highest value.

Table 2. Data set used in these experiments

	Training	Testing
Users	11,921	11,495
EOIs	360,498	560,595
Positive Replies	56,080	93,810
Baseline Success Rate	(15.56%)	(16.73%)
Negative Replies	164,880	309,211
Baseline Failure Rate	(45.74%)	(55.16%)

and negative preference models; that is, users who have sent at least one EOI or replied positively to one EOI (a positive indication of preference), *and* who have replied negatively to at least one other user. In order to run different types of experiments in a timely manner and because location is one of most important factors in online dating, we only selected users who lived in the Sydney area. The size of the data set is shown in Table 2. The training and test sets are similar in size. The baseline success rate of is calculated by dividing the positive replies by the total number of EOIs. Similarly, the baseline failure rate is the number of negative replies as a percentage of all EOIs.

4.1 Evaluation Metrics

We evaluated our systems using EOI precision at N ($P@N$), success rate at N ($S@N$), and failure rate at N ($F@N$). EOI precision at N (Equation 3), measures the proportion of the top-N recommendations to whom the user sent an EOI in the test data. EOI precision at N can tell us how well the ranking works in terms of the rate of acceptance of the recommendations by the user receiving the recommendation.

$$P@N = \frac{|EOIs \cap Recommended|}{|Recommended|} \tag{3}$$

Success rate at N ($S@N$) measures the rate of success (EOI with positive replies) among all EOIs in the top-N recommendations (Equation 4). Success rate at N can tell us whether the first N recommendations, if accepted by the users receiving them, are likely to have positive responses.

$$S@N = \frac{|EOIs \cap Recommended \cap PositiveResponse|}{|EOIs \cap Recommended|} \tag{4}$$

Failure rate at N is a similar measure to $S@N$, and is calculated using all EOIs that had negative responses against all EOIs in the top-N recommendations (Equation 5). Failure rate can tell us whether a ranking strategy can help minimise negative responses if the first N recommendation were accepted by the users. Therefore, $F@N$ is particularly important for evaluating a strategy which aims to minimise user dissatisfaction. Note that an EOIs can have a positive response, a negative response or they may have no response at all; so $F@N$ is not the complement of $S@N$.

$$F@N = \frac{|EOIs \cap Recommended \cap NegativeResponse|}{|EOIs \cap Recommended|} \quad (5)$$

4.2 Results

We analyse how well the compatibility scores correlate with the users actual responses to each other by observing if higher positive compatibility $C^{+}(A, B)$ between all users A and B translate into more EOIs sent between users A and B.

We observed that all compatibility scores are normally distributed across the number of EOIs sent. We also noticed a higher average positive compatibility score (average: 0.45, standard deviation: 0.07) than the average negative compatibility score (average: 0.40, standard deviation: 0.09). Most EOIs are sent between users with combined compatibility higher than 0.5, meaning that their positive scores are higher than their negative scores. The average combined score is 0.51 with a standard deviation of 0.07, meaning that the positive compatibility scores between the sender and receiver of EOIs is mostly higher than their negative compatibility score. The higher standard deviation for the negative compatibility scores is likely due to the user's lack of control over who sends them an EOI. This means that people receive messages from users with a wider ranger of attributes, compared to the range of attributes found in the positive preference models.

In order to understand how negative models of preference can help avoiding undesired recommendations, we used a set of users who have both a positive preference model (have sent at least one EOI) and a negative preference model (have sent at least one negative reply). For these users, we observed no significant difference in $P@N$ when including the combining positive and negative preferences in comparison to positive preferences only as shown in Figure 1. The few EOI-precision points that we are losing are recommendations with high positive compatibility scores and with similarly high negative compatibility scores. As shown in example of Table 1 with recommendations p, a highly positive recommendation based on $C^{+}(x, p)$ are pushed down the ranking on $C^{\pm}(x, p)$, because the negative compatibility score $C^{-}(x, p)$ is equally high.

It is important to highlight that because we are evaluating over historical data, the values of $P@N$ are lowerbound values. For instance, from all recommendations that we generate, we are only certain of those that are present in the historical data, most recommendations were not seen by the user and therefore nothing can be inferred for those. Therefore, if we generate 100 recommendations and only 5 appear in the historical data, we can say for certain that we have a 5% lowerbound precision, but we cannot say anything regarding the remaining 95 recommendations.

Because the negative preference model of a user *Alice* is based on the few users who sent EOIs to her and to whom she sent negative responses, the ma-

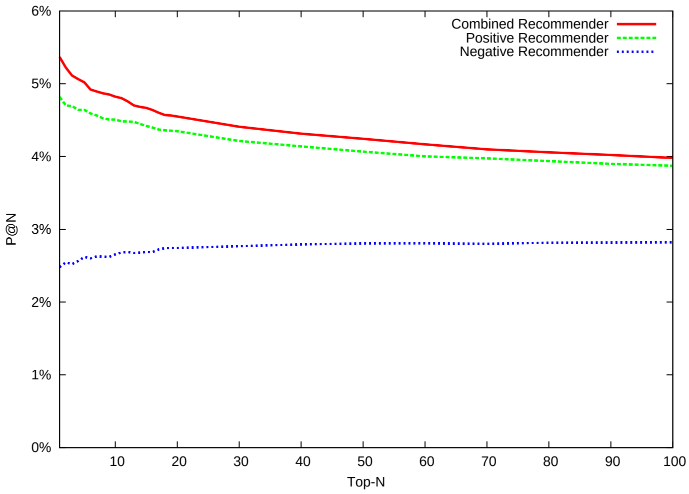

Fig. 1. EOI precision for different number of recommendations given.

jority of the online users will not match *Alice*'s negative preference model. For instance, *Alice* may not want to date users over 50 years of age, but because she has not receive any EOI from users of this age group (and as a consequence, she has not rejected them), we cannot infer a negative compatibility score for them. Therefore, most random pairs of online dating users will have unknown negative compatibility scores. In our strategy unknown scores are given the value of zero (meaning *not disliked*). Because of this, we evaluated the negative preference model scores using all recommendations B where the positive compatibility $C^{+}(A, B)$ is greater than zero (i.e. users we predict A likes to at least some degree). For this recommender – referred to as 'Negative Recommender' – we ranked all recommendations such that the recommendation B with the lowest negative compatibility score $C^{-}(A, B)$ (least disliked) appears first and the B with highest negative compatibility score appears last.

We observed in Figure 1 that the 'Negative recommender' seems to have constant EOI precision, which indicates that the negative compatibility scores by itself does not provide a good ranking for a recommender. Also, we observed that many users have low negative compatibility scores, which indicate that this model contains many ties, which will harm the precision of such a recommender. Another reason why negative compatibility scores cannot predict EOIs is that the negative compatibility model is trained over responses of EOIs and not EOIs that were not sent (information that we do not possess). For the same reason, the combined recommender has similar EOI precision to the positive recommender.

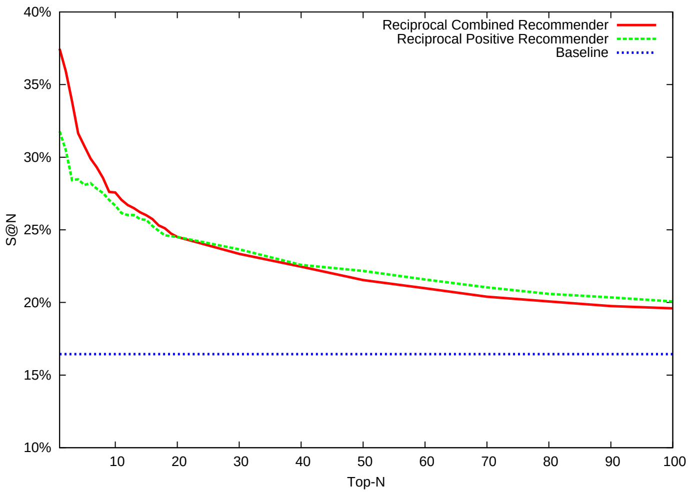

Fig. 2. Success rate for different number of recommendations given.

Unlike other recommendations, reciprocal recommendations benefit from negative preferences as can be seen in Figure 2. This occurs because negative preferences are modelled using negative *responses*, therefore improving measures that account for the response of the users. We can observe that the use of negative preferences in the reciprocal recommender gives a better success rate for top-1 and for top-5, while for top-10 and for top-100 they are virtually the same. The results for the reciprocal combined recommender for top-1 and for top-5 are 37.46% and 30.77% respectively, while the corresponding results for the reciprocal positive-only recommender are 31.78% and 28.09%. We ran the Mann-Whitney-Wilcoxon test on both data sets, for different values of N, to see whether the success rate improvement of the negative preference reciprocal recommender was statistically significant. We found that the difference is significant to a 95% confidence interval from top-1 to top-5 but is not significant at higher N. Importantly, the success rates of both recommenders are higher than the baseline success rate, which is the ratio between the number of positively replied EOIs and the number of EOIs in the test set. These results are important for our domain, particularly for the case of unpopular users, for whom we may have small numbers of good recommendations. These results are also important as the very top recommendations are critical because people are most likely to focus time and attention on the first set of items presented to them [6].

Matching these results for success rate, there is a lower failure rate for reciprocal preferences when negative preferences are used, compared to the case when only positive preferences are used (Figure 3). We can observe that for all values

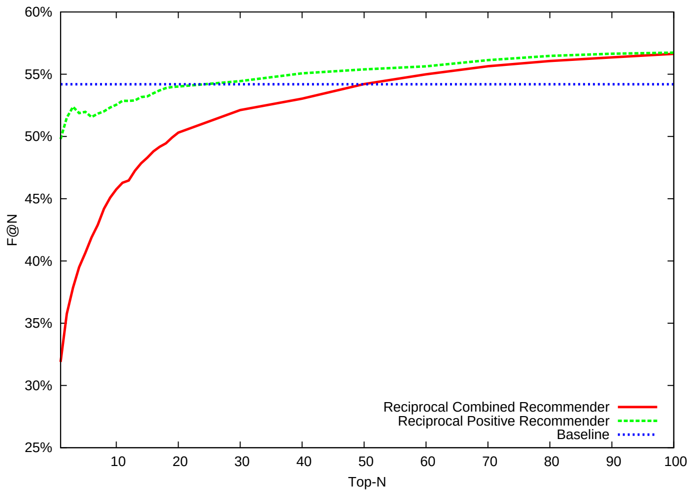

Fig. 3. Failure rate for different number of recommendations given.

of N the combined positive and negative recommender consistently outperforms the positive only recommender. But at top-100, the failure rate increases above the baseline level of 54.19%, which indicates that the reciprocal recommender provides recommendations with lower chance of rejection only for lower numbers of N.

Overall these results show that exploiting negative preferences is a promising approach for fine tuning reciprocal recommendations. As the negative preference compatibility score is subtracted from the positive one, the effect is in fact only pushing recommended users down the list, not promoting users to be recommended who did not already have a very high positive score. This is why the EOI precision is not statistically different. However for top-1 to top-5, the success rate is higher and the failure rate considerably lower than using the positive preferences only. This means that our combined recommender does indeed help with our goals of reducing the chance of rejection.

5 Concluding Remarks

The driver for this exploration of modelling both negative and positive preferences was to reduce the risk of people being rejected in a reciprocal recommender system. Our broader goal was to gain greater understanding of how the negative preference model affects the performance of a reciprocal recommender. Accordingly, we defined models for both positive and negative preferences and explored ways to combine these to select recommendations and then for ranking them.

We have conducted our research in the context of a large online dating site. In our study, we created a model of negative user preferences and evaluated the use of this model in conjunction with the use of a positive model of user preferences in order to generate and to rank recommendations for an online dating recommender. We observed that, despite the fact that negative preferences do not help to increase the number of EOI sent (observed using $P@N$), they do help to make recommendations with a higher chance of success and lower chance of failure. Therefore, by accounting for *dislikes* as well as *likes*, the addition of negative preferences in a reciprocal recommender can reduce the risk of repeated rejection that some users experience in online dating. Since other reciprocal recommender domains also involve a risk of rejection, these results are a contribution to improving understanding of how to create better reciprocal recommenders.

Acknowledgments

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

References

1. Akehurst, J., Koprinska, I., Yacef, K., Pizzato, L., Kay, J., Rej, T.: CCR - A Content-Collaborative Reciprocal Recommender for Online Dating. In: Proc. of the 22nd International Joint Conference on Artificial Intelligence (IJCAI). Barcelona (2011)
2. Broovsk, L., Petek, V.: Recommender system for online dating service. CoRR abs/cs/0703042 (2007)
3. Cai, X., Bain, M., Krzywicki, A., Wobcke, W., Kim, Y., Compton, P., Mahidadia, A.: Collaborative filtering for people to people recommendation in social networks. In: Advances in Artificial Intelligence, LNCS, vol. 6464, pp. 476–485. Springer (2011)
4. Chao, D.L., Balthrop, J., Forrest, S.: Adaptive radio: achieving consensus using negative preferences. In: Proc. of the 2005 international ACM SIGGROUP conference on Supporting group work. pp. 120–123. GROUP '05, ACM (2005)
5. Diaz, F., Metzler, D., Amer-Yahia, S.: Relevance and ranking in online dating systems. In: SIGIR '10: Proc. of the 33rd international ACM SIGIR conference on Research and development in information retrieval. pp. 66–73. ACM (2010)
6. Joachims, T., Granka, L., Pan, B., Hembrooke, H., Gay, G.: Accurately interpreting clickthrough data as implicit feedback. In: Proc. of the 28th annual international ACM SIGIR conference on Research and development in information retrieval. pp. 154–161. SIGIR '05, ACM, New York, NY, USA (2005)
7. Kim, Y., Mahidadia, A., Compton, P., Cai, X., Bain, M., Krzywicki, A., Wobcke, W.: People recommendation based on aggregated bidirectional intentions in social network site. In: Knowledge Management and Acquisition for Smart Systems and Services, LNCS, vol. 6232, pp. 247–260. Springer (2010)
8. McFee, B., Lanckriet, G.: Metric learning to rank. In: Proc. of the 27th International Conference on Machine Learning (ICML'10) (June 2010)
9. Pizzato, L., Rej, T., Chung, T., Koprinska, I., Kay, J.: Recon: a reciprocal recommender for online dating. In: RecSys '10: Proc. of the fourth ACM conference on Recommender systems. pp. 207–214. ACM, New York, NY, USA (2010)