CCR - A Content-Collaborative Reciprocal Recommender for Online Dating

Joshua Akehurst, Irena Koprinska, Kalina Yacef, Luiz Pizzato, Judy Kay and Tomasz Rej
School of Information Technologies
University of Sydney, Sydney, Australia
{firstname.lastname}@sydney.edu.au

Abstract

We present a new recommender system for online dating. Using a large dataset from a major online dating website, we first show that similar people, as defined by a set of personal attributes, like and dislike similar people and are liked and disliked by similar people. This analysis provides the foundation for our Content-Collaborative Reciprocal (CCR) recommender approach. The content-based part uses selected user profile features and similarity measure to generate a set of similar users. The collaborative filtering part uses the interactions of the similar users, including the people they like/dislike and are liked/disliked by, to produce reciprocal recommendations. CCR addresses the cold start problem of new users joining the site by being able to provide recommendations immediately, based on their profiles. Evaluation results show that the success rate of the recommendations is 69.26% compared with a baseline of 35.19% for the top 10 ranked recommendations.

1 Introduction

Recommender systems have been the focus of much research in recent years, both in academia and in the commercial world, providing people with recommendations for books, music and movies among many other things. There are two major approaches for generating recommendations, content-based and collaborative filtering, along with a variety of methods to combine them [Burke, 2002; Degemmis *et al*., 2007]. While most systems recommend items to people, recommending people to other people has begun to emerge under the names of social matching systems [Terveen and McDonald, 2005] and reciprocal recommenders [Pizzato *et al*., 2010a]. This class of recommenders differs from the traditional items to people recommenders. In a reciprocal recommender both sides can express their likes and dislikes and a good match requires satisfying the preferences of both people; in traditional recommenders, it requires only satisfying the preferences of the person for whom the recommendations are generated.

Our paper focuses on the reciprocal domain of online dating. This is a new research area, with only a few recently published papers. Cai *et al.* [2010] introduced a collaborative filtering algorithm based on user similarity in taste and attractiveness. Pizzato *et al.* [2010a] proposed a content-based approach and showed that taking into account reciprocity improves recommendations. Diaz *et al.* [2010] proposed to learn a ranking function that maximizes the number of positive interactions between online dating users based on user profiles.

In this paper we propose a hybrid content-collaborative recommender for online dating that uses both user profiles and user interactions. Our contributions can be summarized as follows:

- We investigate the hypothesis that people with similar profiles like and dislike similar people, and are liked and disliked by similar people. We use the results as the foundation for our recommender system.
- We propose CCR, a new content-collaborative reciprocal recommender. To recommend people to a given user, the content-based part uses similarity between user profiles to find users who are similar to the target user. The collaborative filtering part then uses the interaction history of the similar users, including the people they like/dislike and are liked/disliked by, to produce the final recommendations.
- As part of our recommender, we propose an efficient double-constraint algorithm for generating similar users and candidates, and a novel approach for candidate ranking based on user interactions. The later is supported by [Pizzato *et al*., 2010b; Diaz *et al.,* 2010] who found that the user preferences inferred from the interactions with other users are more reliable than the explicitly stated user preferences.
- We evaluate CCR using a large dataset from a popular online dating website.

An important advantage of CCR is that it avoids the cold start problem. New users are not required to provide any input apart from their profile before receiving recommendations and are therefore able to engage with the system right from the start.

This paper is organized as follows. Section 2 describes the user interaction on the dating website. Section 3 presents the analysis of similar users. Section 4 explains the proposed recommendation approach. Section 5 describes the evaluation setup and Section 6 presents and discusses the results. Section 7 concludes the paper.

2 Domain Overview

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

1) Creating a *user profile* – New users login to the web site and provide information about themselves such as their age, gender, location, education, personality type, diet, drinks, religion and interests.

2) Browsing the 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* and that the EOI was *successful.*

4) Unmediated interaction – *A* or *B* buy tokens from the website (usually after a successful EOI) to send each other unmediated message. This is the only way to exchange contact details.

3 Correlation Analysis: Do Similar People Like/Dislike Similar People and are Liked/Disliked by Similar People?

We test these hypotheses for the case when the similarity is based on user profiles.

Being able to automatically generate a set of recommendations that a user may *like*, based on his/her profile is very useful for traditional content-based recommenders and addresses the cold start problem. However, it is not enough for reciprocal recommenders; that is why we also explore whether similar people *dislike* similar people, are *liked* by, *disliked by* and *reciprocally liked* by similar people.

3.1 Interaction Groups

For a user *U*, we define five *interaction groups* of users*: U Likes*, *U is Liked By*, *U Dislikes*, *U is Disliked By* and *U is Reciprocally Liked By*. These groups represent the interaction of *U* with the other users and are constructed based on the EOIs sent and received by *U* and their responses:

- *U Likes*: The set of users that we know *U* is interested in. It contains all users *Y* to whom *U* has sent an EOI, regardless of the response, as well as all users *X* to whom *U* has responded positively after they sent *U* an EOI, Figure 1a.
- *U is Liked By*: The set of users that we know are interested in *U*. It contains all users *X* who sent an EOI to *U*, regardless of the response, as well as all users *Y* who responded positively to an EOI from *U*, Figure 1b.
- *U Dislikes*: The set of users that we know *U* is not interested in. It contains all users *X* to whom *U* responded negatively after receiving an EOI from them, Figure 1c.
- *U is Disliked By*: The set of users that we know are not interested in *U*. It contains all users *Y* who have responded negatively to *U* after *U* sent them an EOI, Figure 1d.
- *U is Reciprocally Liked By:* The set of users that we know *U* is interested in who are also interested in *U*. This is the intersection of *U Likes* and *U is Liked By*. It is the set of users *Y* to whom *U* sent an EOI where the response was positive, as well as the set of users *X* who sent an EOI to *U* where *U* has responded positively, Figure1e.

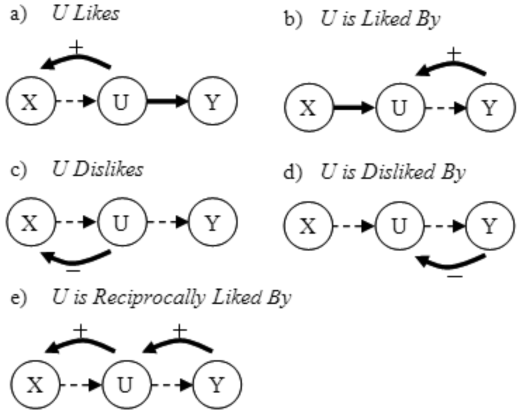


Figure 1: Interaction groups for user *U*. The + and - arcs indicate positive and negative responses to EOIs; in bold are the arcs that apply in each case.

3.2 Data-Preprocessing and Selected Attributes

Our hypothesis was evaluated using real data from a popular online dating website. The dataset consists of anonymised user profile attributes and information about the interactions between these users, i.e. EOIs sent and responses received. Due to the size of the dataset we only selected users residing in a single geographic region. We then removed users who had not sent or received at least one EOI within a one month period (March 2010). Table 1 summarizes the data statistics.

Total users	216,662
Male users	119,102 (54.97%)
Female users	97,560 (45.03%)
EOIs	167,810
Successful EOIs	24,079 (25.59%)
Users with at least 1 EOI	7,322
Male users with at least 1 EOI	3,965
Female users with at least 1 EOI	3,357

Table 1: Data statistics

The original dataset consists of 39 user profile attributes. We conducted a preliminary data analysis of the distribution of these attributes to identify both the importance and suitability of each attribute. Using this analysis and after some trials of computing correlations, we manually selected a subset of 7 attributes. In addition, some attribute values were merged together, e.g. the values *overweight* and *largish* of *body_type* were merged into *overweight*. Table 2 shows the resulting user profile attributes and their values.

Attribute	Type	Possible Values
age	numeric	18-81
height	numeric	149-194
body type	nominal	slim, average, overweight
education level	nominal	secondary, technical, university
smoker	nominal	yes, no
have children	nominal	yes, no
marital status	nominal	single, previously married

Table 2: Selected profile attributes

3.3 Distance Metric

We define the distance between the profiles of users *A* and *B* as the sum of the three components, the distance between: their nominal attributes (*D_nom*), their *age* (*D_age*) and their *height* attributes (*D_height*):

D(A,B)=D_nom+2D_age+D_height.

To compute *D_nom*, the nominal attributes are first converted into reflected binary representation (Gray code). In this encoding the adjacent values of an attribute have a Hamming difference of 1; e.g. *body_type=slim* is encoded as 00, *body_type=average* as 01 and *body_type=overweight* as 11. *D_nom* is then the Hamming distance between the binary vectors of *A* and *B*.

To measure how similar *A* and *B* are in age and height, after an analysis of the value distribution, we devised the following method. We first compute the absolute difference between the two values and then *D_age* and *D_height* as:

- 0: if the absolute difference is 0-5 units
- 1: if the absolute difference is 5-10 units
- 2: if the absolute difference is greater than 10 units

In addition, *D_age* is weighted by a factor of 2 to account for the importance of the age difference. This weighting was set empirically after data analysis.

For example, given *A* = [18yrs, 180cm, 0110110011] and *B* = [25yrs, 176cm, 1011111100], the difference between their user profiles *D(A,B)* will be:

$$D(A,B) = D_H(0110110011\,,1011111100) + \\ + 2f(|18-25|) + f(|180-176)| = 7+2+0 = 9$$

The difference between the profiles of two users can take values from 0 to 16, with 0 indicating maximum similarity and 16 indicating maximum dissimilarity.

3.4 Method

Using the distance metric and interaction sets defined above, we test our hypothesis if similar people like/dislike similar people and are liked/disliked by similar people.

Given a pair of users *A* and *B*, we compute the correlation of the similarity between *A* and *B* and the similarity between their interaction groups. The similarity between two interaction groups is computed as the similarity between the group centroids. More specifically, we compute *i*, the distance between each pair of users, and *j,* the distance between the centroids of the corresponding interaction groups, as shown in Figure 2, and then compute the correlation between *i* and *j*. This was repeated for males and females separately and together. Baseline correlations were computed by randomly selecting users of the same gender, with an age difference of up to 5 years, to fill in the interaction group; the number of users was the same as in the original interaction group.

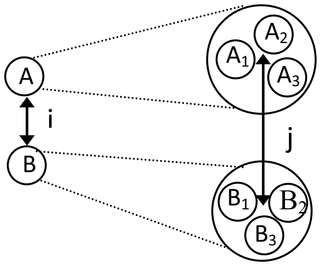


Figure 2: Correlation analysis

Table 3 shows the Pearson correlation values averaged over all users; all of them are statistically significant at p<0.01 level. The values are in the range of [0.439-0.514] for females, [0.404-0.545] for males and [0.424-0.529] for females and males together. It is important to note that correlation values around 0.5 are considered high in studies involving human subjects [Coon and Mitterer, 2007].

Another important observation is that all correlation values are considerably higher than the respective baseline values (by a factor of 10-30), and that these differences are statistically significant in all cases at p<0.01 level. Thus, the selected features and distance metric provide a huge gain over the baseline.

It is also interesting to note that in most cases the correlation values are slightly lower for females than for males. This is expected as females are known to be more selective in their EOIs and less likely to act in unison [Trivers, 2009].

	Computed			Baseline		
	F	M	F+M	F	M	F+M
U Likes	0.496	0.545	0.524	0.031	0.039	0.038
U Dislikes	0.447	0.404	0.425	0.021	0.015	0.018
U is Liked By	0.514	0.544	0.529	0.070	0.032	0.044
U is Disliked By	0.439	0.527	0.497	0.019	0.028	0.027
U is Reciprocally Liked By	0.462	0.519	0.493	0.031	0.052	0.039

Table 3: Correlation between similar users and their five action sets (F – female, M – male, F+M – male and female combined)

In summary, the results show that in general, similar people like and dislike similar people, and are liked and disliked by similar people. Given user profile, we can generate with a relative confidence a set of users that the user may like, dislike, be liked by or disliked by. This will be the basis for our recommender and will be fundamental for addressing the cold start problem of new users joining the site.

4 CCR – A Hybrid Content-Collaborative Reciprocal Recommender

The two most common recommendation techniques are content-based and collaborative filtering. Content-based approaches [Pazzani and Billsus, 2007] build profiles of user interests based on user stated preferences (e.g. thriller movies) and/or the features of the items rated by the user, and use this model to recommend new items that have similar features. These approaches traditionally do not rely on data from other users. Collaborative approaches [Schafer *et al.*, 2007], in contrast, exploit data from other users who gave similar ratings on the same items and recommend items that have been rated positively by other users and not yet by the target user. Both approaches have their advantages and drawbacks; hybrid recommenders combine them to address the limitations of each approach [Burke, 2002; Degemmis *et al.*, 2007]. Our domain of online dating poses new challenges as the recommended items are people, with their own preferences, profile and activity history. Further, in domains such as online dating it is crucial to provide useful recommendations to new users, so that they are not discouraged at the beginning, hence making the cold-start problem critical to solve. We propose CCR, a hybrid recommender that addresses the notion of reciprocity as well as the cold start problem.

4.1 Rationale

Our recommender approach computes the similarities between users relying on the content of their profiles and then produces recommendations that are likely to be reciprocally successful, based on the likes and dislikes of similar users.

The correlation results suggest that if two users are similar, as defined by our distance metric, then they will like and be liked by similar groups of users, as well as dislike and be disliked by similar groups of users with a reasonable degree of certainty. They will also have reciprocal interest with similar groups of users.

Exploiting these results, we use a collaborative filtering technique to generate recommendations to a new user *U* and rank them by the likelihood of reciprocal taste. The CCR algorithm is presented in the next section.

4.2 Algorithm

The process of generating recommendations (an ordered list of users) for a given user *U* comprises of three key steps, see Figure 3. First, we find a set of users S_u that are similar to *U* using the selected profile attributes and distance metric. Second, we examine the interactions of the users in S_u and produce a set of candidate recommendations C_u. Third, we rank the candidates. We discuss these steps below.

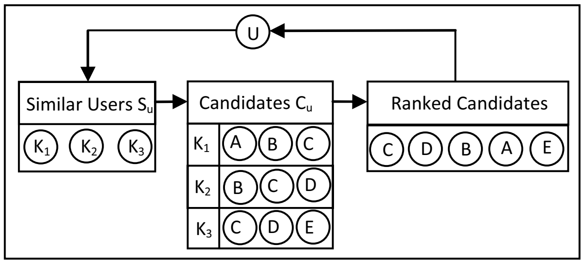


Figure 3: Recommender process

Generating Similar Users Based on User Profile

Given a user *U*, the aim of this step is to produce the set of similar users *Su*, containing *K* users who have the lowest possible distances to *U*. For example, Figure 3 shows that $S_u=\{K_1, K_2, K_3\}$. The size of S_u is bounded by a double constraint: at least the *K* nearest users are acquired and the size of the candidate list C_u is at least of size *C*. This ensures that users who contribute a high number of candidates will not dominate the candidate list and that there will be a sufficiently large number of possible candidates from which to generate recommendations.

Generating Candidates Based on User Interaction

Once S_u has been generated, we produce an unordered list of candidate recommendations C_u. More specifically, for every user in S_u, we retrieve the list of all users that he/she has reciprocal interest with. These users become candidates for recommendation and are included in C_u. For example, in Figure 2 the set of users with reciprocal interest for K_1 is *{A,B,C}*, for K_2: *{B, C, D}*, for K_3: *{C, D, E}* and the recommendation candidate set for *A* is the union of these sets, $C_A=\{A, B, C, D, E\}$.

Generating a Ranked List of Recommendations

We then compute the level of support for each candidate based on the interactions between the similar users S_u and the candidate pool C_u. Users are added to C_u if they have responded positively to at least one S_u user or have received a positive reply from at least one S_u user. However, some candidates might have received an EOI from more than one *Su* user and responded to some positively and to others negatively. Thus, some candidates have more successful interactions with S_u than others. Our ranking method, called Support, computes the support of S_u for each candidate.

For each candidate *X* we calculate the number of times *X* has responded positively or has received a positive response from S_u, see Table 4. We also calculate the number of times *X* has responded negatively or has received a negative response from S_u. The support score for *X* is the number of positive minus the number of negative interactions; we found this simple method to work best. The higher the score for *X*, the more reciprocally liked is *X* by S_u. The candidates are sorted in descending order based on their support score.

X	# Positive responses		# Negative responses		Score
	X->S_u	S_u->X	X->S_u	S_u->X	
A	2	0	6	1	-5
B	4	2	4	1	1
C	10	2	4	2	6
D	5	1	1	1	4

Table 4: Support ranking example. The ranking will be C, D, B, A.

This ranking method ensures that candidates who are reciprocally liked by more users in S_u are ranked higher than candidates who are less liked. At the same time it also accounts for the over-representation of popular candidates as they are more likely to have a higher negative response rate.

5 Experimental Evaluation

5.1 Data

We used the data as shown in Table 1. For each run of the experiment, the dataset is partitioned into two distinct sets, training and testing, containing approximately 2/3 and 1/3 of the users, respectively. Each training/testing partition contains an even distribution of males and females. Each set was also evenly assigned users who were more popular than their cohort in terms of EOI sent and/or received.

EOIs and their responses from users in the training set to users in the testing set and vice versa were removed to ensure that the testing set is not polluted by the training set. Users in either the testing or training set who no longer meet the minimum number of EOI required were removed. These processing resulted in the removal of less than 1% of the users before the segmentation into training and test sets. Information about the interactions of the users from the testing set is never included when ranking the candidates for this user to ensure clear separation between the two sets.

5.2 Generating S_u and C_u

We start with a maximum distance threshold of 0. For a given user *U* from the test set, we consider the users from the training set in random order and compute the distance between them and *U;* if the distance is below the threshold, these users are added to S_u. Once all of the users in the training set have been considered, the threshold increases by 1 (and the distance results are cashed so that comparisons are not repeated). This process continues until at least *K* users have been added to S_u and the size of C_u is at least C. This process is quite efficient on average as most users satisfy the double constraint at low thresholds, especially when the training sets contain a large number of users.

5.3 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 and the response was positive.
- Unsuccessful EOI sent by *U, unsuccessful_sent*: The set of users who *U* has sent an EOI and the response was negative.
- 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 and unsuccessful EOI for *U*:
  successful = *successful_sent* + *successful_recv*
  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 this set as:

successful@N = *successful* ∩ *N_recommendations*
unsuccessful@N = *unsuccessful* ∩ *N_recommendations*

Then, given a set of *N* ordered recommendations, the *success rate @ N* is the number of correct recommendations over the number of all interacted recommendations:

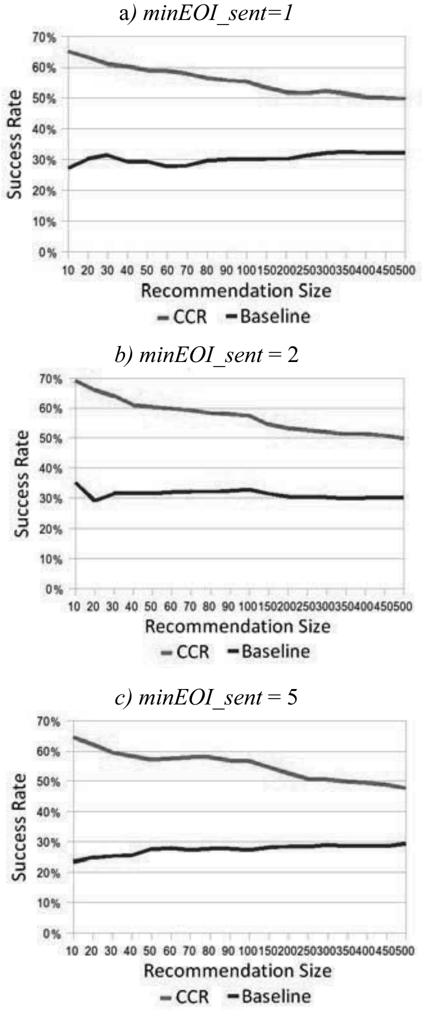
$$\text{success rate}@N = \frac{\#\,successful\,@\,N}{\#\,successful\,@\,N + \#\,unsuccessful\,@\,N}$$

For comparison we use the following baseline: the success rate of the recommender using a random set of *K* users in S_u as opposed to *K* nearest neighbors.

The success rate is calculated for each user in the test set and the overall success rate is the average.

6 Results and Discussion

In each evaluation the top *N* recommendations were tested. We evaluated the effect of *N* and the minimum number of EOI sent by a user (*minEOI_sent*) by varying *N* from 10 to 500 and *EOI_sent* from 1 to 20. For brevity we present the results for *EOI_sent* =1, 2 and 5; the number of users in the testing set for these cases was 7,322, 5,709 and 3,710, respectively. Each experiment has been run 10 times and the reported success rate is the average.


a*) minEOI_sent=1*

b) minEOI_sent = 2

c) minEOI_sent = 5

Figure 4: CCR success rate results for various *N* and *minEOI_sent*

Figure 4 shows the success rate results. The CCR recommender significantly outperforms the baseline in all cases. For example, for *N*=10, the success rate of CCR is: 65.81% (baseline=27.69%) for *minEOI_sent*=1; 69.26% (baseline=35.19%) for *minEOI_sent*=2 and 64.49% (baseline=23.44%) for *minEOI_sent*=5.

As the number of recommendations *N* increases from 10 to 500, the success rate decreases by 10-20%. This means that the best recommendations are at the top of the list and adding more recommendations only dilutes the success rate. Hence, our ranking criterion is useful and effective.

In practice, the success rate for a smaller *N*, e.g. *N*=10-30 is very important as this is the typical *N* presented to the user. Unsuccessful recommendations, especially recommendations leading to rejection can be very discouraging.

Our results also show that as the number of *minEOI_sent* increases from 1 to 20, the success rate trends are very similar. However, for users who sent more EOIs (not shown in Figure 4), the success rate is slightly lower (e.g. 60.16% for *minEOI_sent*=10 and 58.54% *minEOI_sent*=20, for *N*=10). This can be explained with the fact that the highly active users may be less selective.

In all experiments we used *K=100* and *C=250.* With these parameters it took approximately 100 milliseconds to generate the recommendation list for a user which confirms the efficiency of our algorithm for generation of similar users and candidate recommendations.

7 Conclusions

This paper presents CCR, a novel reciprocal recommender system for an online dating website. We first conducted a correlation-based data analysis and found that people with similar profiles like and dislike similar people and are liked and disliked by similar people. This provided the foundation for our novel hybrid recommendation approach, which combines content-based and collaborative filtering, and utilizes data from both user profiles and user interactions.

To generate a recommendation for a new user, CCR uses similarity of profile features to retrieve a set of similar users. It then utilises the user interactions of the similar users to generate an ordered list of reciprocal recommendations. It uses an efficient algorithm for generation of similar users and candidate recommendations and also a novel ranking method based on reciprocal interest of the candidates and the similar users.

CCR was evaluated on a large dataset from a major online dating site, over a period of one month. The results showed a significant improvement over the baseline; the success rate of the reciprocal recommendations was [64.24 - 69.26%] for different number of EOI, compared with success rate baseline of [23.44 - 35.19%]. We also found the ranking method to be effective, with the most successful recommendations appearing at the top of the list.

An important advantage of CCR is that it addresses the cold start problem of new users joining the website by being able to provide recommendations immediately, based on the profile of the new user. This is very important for engaging the new users.

The results of this study are relevant not only for online dating but also for other reciprocal domains such as matching employees with employers and mentors with mentees.

Future work will investigate better combinations of features, feature weighting and distance metrics. It will also consider explicitly constraining some attributes, for example recommending taller and older men to women.

Acknowledgments

This work was supported by the Smart Services Cooperative Research Centre.

References

[Burke, 2002] R. Burke. Hybrid recommender systems: survey and experiments. *User Modeling and User-Adapted Interaction* 12:331-370, 2002.

[Cai *et al.*] X. Cai, M. Bain, A. Krzywicki, W. Wobcke, Y.S. Kim, P. Compton, A. Mahidadia. Collaborative filtering for people to people recommendation in social networks. In *Proc. Australian Joint Conference on Artificial Intelligence*. LNCS 6464:476-485, 2010. Springer.

[Coon and Mitterer, 2007] D. Coon and J. Mitterer. *Introduction to Psychology*. Thomson, 2007.

[Degemmis *et al.*, 2007] M. Degemmis, P. Lops, and G. Semeraro. A content-collaborative recommender that exploits WordNet-based user profiles for neighborhood formation. *User Modeling and User-Adapted Interaction* 17:217-255, 2007.

[Diaz *et al.*, 2010] F. Diaz, D. Metzler, and S. Amer-Yahia. Relevance and ranking in online dating systems. In *Proc. SIGIR Conference*, 66-73, 2010. ACM.

[Pazzani and Billsus, 2007] M.J. Pazzani and D. Billsus. Content-based recommendation systems. In *The Adaptive Web*, LNCS 4321:325-341, 2007. Springer.

[Pizzato *et al.*, 2010a] L. Pizzato, T. Rej, T. Chung, I. Koprinska, J. Kay. RECON: a reciprocal recommender for online dating. In *Proc. ACM Conference on Recommender Systems (RecSys),* Barcelona, 2010. ACM.

[Pizzato *et al.*, 2010b] L. Pizzato, T. Chung, T. Rej, I. Koprinska, K. Yacef and J. Kay. Learning user preferences in online dating. In Proc. *Preference Learning Workshop, ECML PKDD Conference*, Barcelona, 2010.

[Schafer *et al.*, 2007] J.B. Schafer, D. Frankowski, J. Herlocker, and S. Sen. Collaborative filtering recommender systems. In *The Adaptive Web*, LNCS 4321:291-324, 2007. Springer.

[Terveen and McDonald, 2005]. L. Terveen and D.W. McDonald. Social matching: a framework and research agenda. *ACM Transactions on Computer-Human Interaction*, 12(3):401-434, 2005.

[Trivers, 2009] R.L. Trivers. Parental investment and sexual selection. In *Sexual Selection and the Descent of Man*, Transaction Publishers.