A Deployed People-to-People Recommender System in Online Dating

*Wayne Wobcke, Alfred Krzywicki, Yang Sok Kim, Xiongcai Cai,
Michael Bain, Paul Compton, and Ashesh Mahidadia*

■ *Online dating is a prime application area for recommender systems, as users face an abundance of choice, must act on limited information, and are participating in a competitive matching market. This article reports on the successful deployment of a people-to-people recommender system on a large commercial online dating site. The deployment was the result of thorough evaluation and an online trial of a number of methods, including profile-based, collaborative filtering and hybrid algorithms. Results taken a few months after deployment show that the recommender system delivered its projected benefits.*

Recommender systems have become important tools helping users to deal with information overload and the abundance of choice. Traditionally these systems have been used to recommend items to users. The work we describe in this article concerns people-to-people recommendation in an online dating context. People-to-people recommendation is different from item-to-people recommendation: interactions between people are two-way, in that an approach initiated by one person to another can be accepted, rejected or ignored. This has important implications for recommendation. Most basic, for each recommendation, there are two points in time at which the user must be satisfied. First, when a recommendation is given, the user must like the proposed candidate (as in item-to-people recommendation), but second, if the user contacts a presented candidate, the user should be satisfied by the candidate's reply (the user prefers a positive reply to a negative reply or no reply at all). Thus a people-to-people recommender system needs to take into account the preferences of proposed candidates, who determine whether an interaction is successful, in addition to those of the user. People-to-people recommenders are thus reciprocal recommenders (Pizzato et al. 2013). Or, putting this from the point of view of the user, a people-to-people recommender system must take into account both a user's taste (whom they find attractive) and their own attractiveness, so the presented candidates will find them attractive in return, and give a positive reply (Cai et al. 2010).

Using online dating sites can be difficult for many people. First, there is the stress of entering or reentering the dating market. Second, there are a variety of dating sites with competing or confusing claims as to their effectiveness in matching people with the perfect partner — a group of psychologists have recently drawn attention to the questionable validity of some of these claims (Finkel et al. 2012). Third, once joining a site, users may have to fill in a lengthy questionnaire or state preferences for their ideal partner, which can be daunting because if the user's objective is to find a long-term partner, they surely know they will have to compromise, but do not know at the outset what compromises they are likely to make. Fourth, as is well known, dating is a type of market, a matching market, where people compete for scarce resources (partners) and are, in turn, a resource sought by others — see the recent user perspective of a labor market economist on the dating market that explores this idea in depth (Oyer 2014). Finally, users are asked to provide a brief personal summary that must differentiate themselves from others on the site and attract interest. It is difficult to stand out from the crowd: many people's summaries appear cliché-ridden and generic, but according to at least one popular account, perhaps the main objective of the summary is simply is to project an optimistic, carefree outlook that creates a positive first impression and stimulates further interest (Webb 2013).

Once onsite, users are immediately overwhelmed with an abundance of choice in potential partners. Moreover, they may also be the subject of intense interest from existing users now that their profile is public (often a site will heavily promote newly joined users). On most sites, users can choose from amongst thousands of others by sifting through galleries of photos and profiles. The most common way for users to find potential contacts is by advanced search (search based on attributes, such as age, location, education, and occupation, and other keywords). Searches typically return many results, and it may be difficult for users to decide whom to contact. As a result, some users contact those whom they find most attractive; however, this is a poor strategy for generating success, as those popular users then become flooded with contacts but can reply positively to only a very small fraction of them.

A basic problem for users is that it is hard for them to estimate their own desirability so as to choose contacts who are likely to respond positively, and similarly, it is impossible for them to know how much competition they face when contacting another person, and hence to gauge their likely chance of success. In addition, after narrowing down searches by obvious criteria, profile information typically provides limited information to discriminate one user from another. This is why a recommender system can provide much help to users of a dating site.

The organization of this article is as follows. In the next section, we outline the basic problems of recommendation in online dating and introduce our key metrics. Then we present our recommendation methods, and discuss a live trial conducted with the aim of selecting one method for deployment on site (Selection of Recommender for Deployment). The Recommender Deployment section contains details of the deployment process for the winning algorithm. A postdeployment evaluation of the method is then provided, followed by lessons learned.

Recommender Systems

Online dating is a prime application area for recommender systems, as users face an abundance of choice, must act on limited information, and are participating in a competitive matching market. Interactions are two way (a user contacts another, and either receives a positive or negative reply, or receives no reply at all), and thus the recommendation of a candidate to a user must take into account the taste and attractiveness of both user and candidate.

Due to the limited amount of user information on online dating sites (and, of course, users can misrepresent themselves to some extent to present themselves in a more favorable light), our recommender system does not aim to provide the perfect match. Rather, we adopt a probabilistic stance and aim to present users with candidates who they will like and that will increase their chances of a successful interaction. Moreover, success is defined narrowly as the user receiving a positive reply to a contact they initiate. The main reason for this restrictive definition is that, for the dating site we were working with, users can make initial contact for no charge by sending a message drawn from a predefined list, and replies (also free and from a predefined list) are predetermined (by the dating site company) as either positive or negative, leaving no room for ambiguity. Thus we have extremely reliable data for this notion of success. Users can initiate paid open communication to others, with or without these initial contacts. However, the dating site company does not have any data on the success of open communications, or any feedback from person-to-person meetings. Thus the typical user interaction sequence we focus on is illustrated in figure 1.

A secondary objective of our recommender system

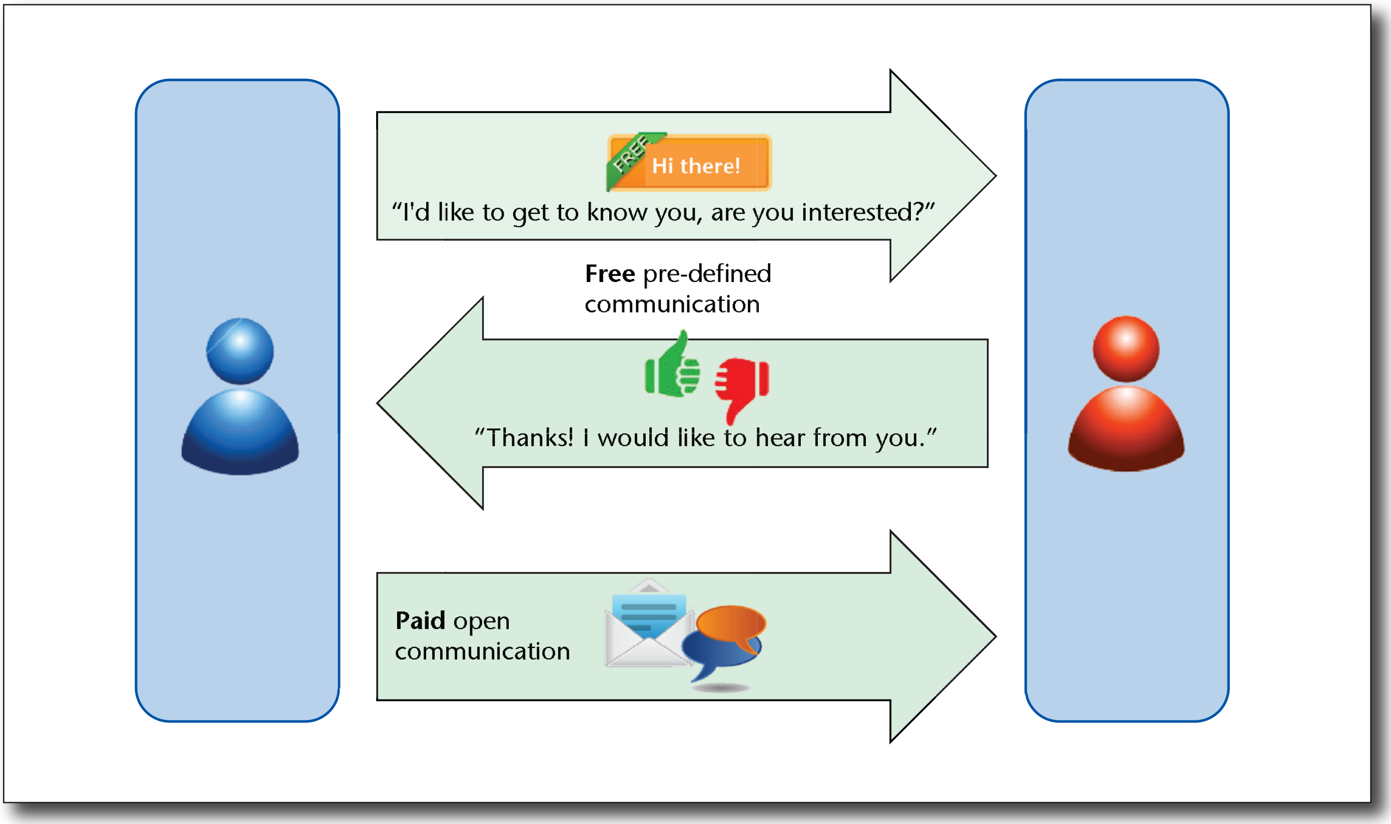
FREE
Hi there!
"I'd like to get to know you, are you interested?"
Free pre-defined communication
"Thanks! I would like to hear from you."
Paid open communication

Figure 1. Typical User Interaction Sequence.

is to increase overall user engagement with the site. If users become frustrated by being unable to find successful interactions, they may leave the site, resulting in an increased attrition rate of users and a reduced candidate pool. By providing opportunities for successful interactions, a recommender system can help maintain a large candidate pool for others to contact. The idea was that this could be achieved by designing the recommender system to promote users who would otherwise not receive much contact, increasing their engagement with the site.

These considerations led us to define two key metrics for the evaluation of recommendation methods: success rate improvement and usage of recommendations. These metrics can apply to an individual, but more commonly we apply them to groups of users. Success rate improvement is a ratio measuring how much more likely users are to have a successful interaction from a recommendation compared to their overall behavior. That is, success rate improvement is a ratio of success rates: the success rate for contacts initiated by users from recommendations divided by the success rate for all contacts of those users. Usage of recommendations measures the proportion of user-initiated contacts that come from recommendations. This is simply the number of user-initiated contacts from recommendations divided by the total number of user-initiated contacts. We also use similar metrics for positive contacts only and for open communications.

It is useful to draw an analogy with information retrieval, where precision and recall are metrics used to evaluate document retrieval methods. Success rate improvement is similar to precision, which measures the proportion of documents out of those presented that are relevant to the user. Recall measures the proportion of the relevant documents that are retrieved by the method. We make two observations. First, recall is not the same as usage of recommendations, since typically the recall metric is applied when every document is known as either relevant or not relevant (so the maximum recall is 100 percent). In a recommendation context, not presenting a candidate is not the same as failing to recall a relevant document, since it is unknown whether a candidate not presented would be of interest to the user. A related issue is that only a very small number of candidates are ever presented to users, not (as in information retrieval) the set of all documents returned by the method. Thus the maximum value for usage of recommendations can never be 100 percent, and in reality is much lower. As a consequence of presenting only very few candidates, the ranking of candidates is highly important: it is much more useful for a recommendation method to generate good results for the top *N* candidates whom users will be shown, rather than being able to produce all candidates of interest. Thus in our historical data analysis, we emphasize success rate improvement and usage of recommendations for the top *N* candidates, with *N* typically in a range from 10 to 100.

Our second observation is that, similar to information retrieval, there is a trade-off between success rate improvement and usage of recommendations. Intuitively, to achieve higher success rate improvement, a method should prefer to generate candidates who say yes more often. However, these candidates are typically not liked as much as other candidates, giving a lower usage of recommendations. Similarly, to achieve higher usage of recommendations, a method should prefer to generate the most attractive candidates, whom users are more certain to like; however, these candidates are more likely to respond negatively or not at all, giving a lower success rate improvement. Thus it is crucial for a recommendation method to find a suitable balance between success rate improvement and usage of recommendations, and much of our research was directed towards finding this balance for a variety of methods. A major difficulty we faced, however, was that it was impossible to determine the right balance from historical data alone, which meant that a live trial was necessary in order to properly evaluate our methods.

In the context of online dating, there are a number of other considerations, which while important, are even more difficult to take into account in the design of a recommender system due to the lack of available data for analysis. One is that the cannibalization of search by recommendation may mean that, while overall user experience might be improved, revenue might not increase because users would spend the same amount of money, but find matches through recommendation rather than search. This seemed to be a general concern with online content delivery prevalent in the media industry. To the contrary, our hypothesis was that an improved user experience would result in more users spending more time on site, and lead to increased revenue. Another consideration is that promoting the engagement of women on the site is important, because even if (some) women do not generate revenue directly, they generate revenue indirectly by constituting the candidate pool for the generally more active men to contact. Assessing the likely performance of a recommender on this criterion was impossible with historical data, and difficult even with data from the trial.

Recommendation Methods

We developed and evaluated a number of recommendation methods that provide ranked recommendations to users that help them increase their chances of success and improve engagement with the site. Our methods are the result of several years of research on people-to-people recommenders starting in 2009, when we developed a number of profile-based and collaborative filtering (CF) methods applied to people-to-people recommendation (Kim et al. 2010, Krzywicki et al. 2010). The most important requirement of our method is scalability: in a typical scenario, the recommender system needed to generate around 100 ranked recommendations for hundreds of thousands of active users by processing several million contacts, using standard hardware in a time period of around 2 hours. As an assessment of standard probabilistic matrix factorization at the time indicated that this method could not scale to data of this size, and moreover could not handle the dynamic and incremental nature of recommendation generation, we focused on simpler methods that would be applicable to problems of this size.

Preliminary evaluation of methods was done on historical data provided by the dating site company, however there was no guarantee that this evaluation would transfer to the setting of a deployed recommender. This is because evaluation on historical data is essentially a prediction task, the objective being to predict the successful interactions that users found by search, that is, in the absence of a recommender. Since a recommender system aims to change user behavior by showing candidates users could perhaps not easily find using search, predicting the positive contacts the user did have is only a partial indication of the quality of the recommendations.

We conducted a first trial of two methods in 2011 over a 9 week period, where recommendations were delivered by email (Krzywicki et al. 2012). The important results of this trial were that: (1) the performance of the methods in the trial setting was consistent with that on historical data, giving us confidence in our methodology, and (2) both methods were able to provide recommendations of consistent quality over a period of time. More precisely, what the first trial showed was that: (1) though success rate improvement and usage of recommendations did not have identical values to those obtained on historical data, broad trends were stable (if one method performed better in the historical setting, it also performed better in the trial setting), and (2) the values of the metrics were roughly similar over different intervals of time within the 9 week trial period.

The CF method used in the first trial did not address the cold start problem (recommendation to and of new users), because we wanted to establish whether a basic form of CF, with similarity based only on common positive contacts, would work in this domain. As the trial results showed that this form of CF worked very well, we developed numerous hybrid CF methods that could recommend to almost all users while maintaining a high success rate (Kim et al. 2012a). This greatly improved the utility of the methods, since providing good recom-

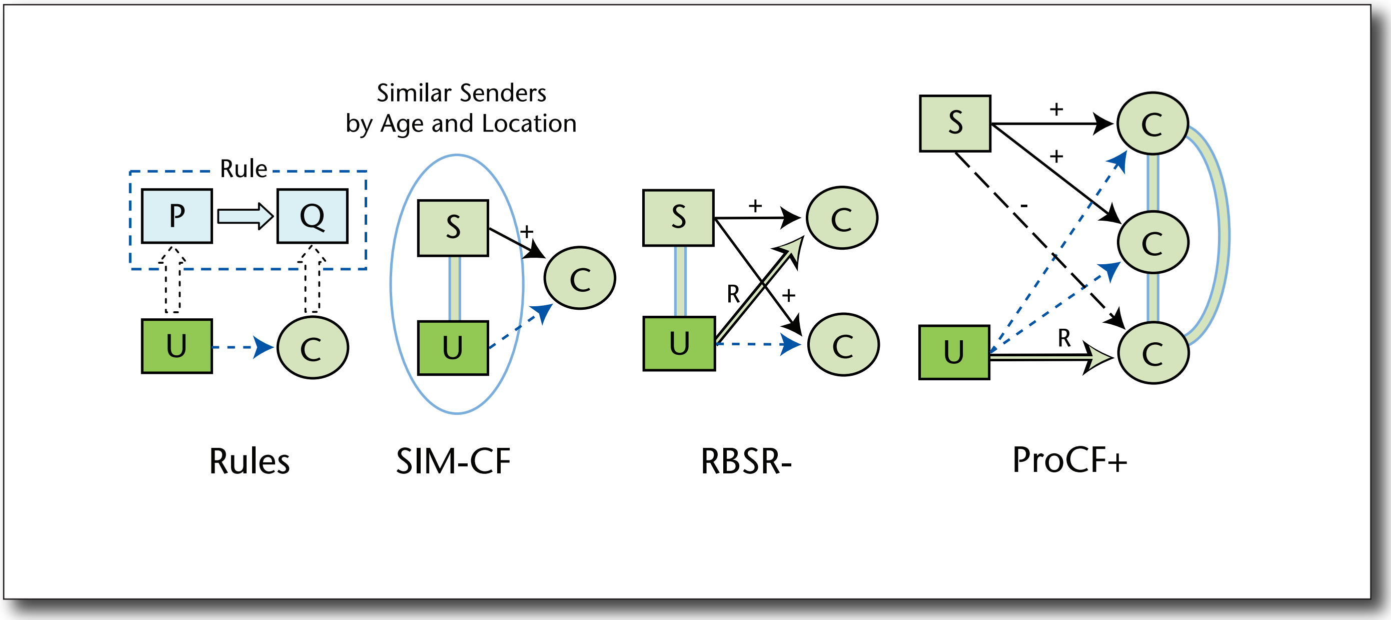

Figure 2. Recommendation Methods.

Candidate *C* can be recommended to user *U* if: (Rules) *C* satisfies the conclusion *Q* of the rule *P* ⇒ *Q* generated for U; (SIM-CF) *C* has replied positively to a user similar in age and location to *U;* (RBSR-CF) *C* has replied positively to a user who initiated a successful interaction with a candidate recommended by Rules to *U;* (ProCF+) *C* is a similar candidate to either a Rules recommendation for *U* or a user who has replied positively to *U,* taking into account successful and unsuccessful interactions with those potential candidates.

mendations to a user on the day of joining the site was considered an effective means of capturing their interest. This article reports on the trial of our best four methods on the same online dating site in 2012, where users were able to click on their recommendations in the browser.

One serious problem that the CF method used in the first trial did address, is the tendency of standard CF to recommend highly popular users. As mentioned above, many users contact the most attractive people, with a low chance of success. More concretely, if we define a popular user as one who has received more than 50 contacts in the previous 28 days, then (1) almost everyone is nonpopular, (2) 38 percent of contacts are to popular users, but (3) the success rate of those contacts is only 11 percent, whereas the success rate of contacts to nonpopular users is 20 percent. Thus there is a serious imbalance in communication patterns that CF-style recommenders tend to reinforce. We addressed this problem by developing a sequential recommendation process, standard CF cascaded (Burke 2002) with a decision tree critic, where the candidates produced by CF are reranked by multiplying their score (an expected success rate improvement derived from their rating) by a weighting less than 1 for those candidates with a strong likelihood of an unsuccessful interaction with the user. The weightings are derived systematically from decision tree rules computed over a large training set of contacts that includes temporal features such as the activity and popularity of users. An effect of this reranking is to demote popular candidates, so that they are not over-recommended. Multiplying scores by rule weights is justified by Bayesian reasoning (Krzywicki et al. 2012).

Summary of Methods

After the first trial, we refined the profile-based matching algorithm, called Rules, and developed three new CF methods, all able to generate candidates for almost all users. The CF methods are of increasing complexity, though all designed to meet the stringent computational requirements imposed by the dating site company, more specifically: (1) be feasible to deploy in commercial settings, (2) be able to generate recommendations in near real time, (3) be scalable in the number of users and contacts, and (4) work with a highly dynamic user base. Our four trialled methods are designed to make various tradeoffs between computational complexity, success rate improvement, usage of recommendations, and diversity of candidates (the number of distinct candidates and the distribution of their recommendations). The methods are summarized in table 1 for reference, and these design trade-offs are discussed further below. Figure 2 shows the generation of candidates in all four methods pictorially. The diagrams do not show the application of the decision tree critic in SIM-CF and RBSR-CF, nor how the candidates are ranked.

Rules	Profile matching method optimizing user coverage and diversity
SIM-CF	Profile-based user similarity with CF, cascaded with decision tree rules that conservatively demote candidates likely to result in unsuccessful interactions
RBSR-CF	Content-boosted CF method exploiting user-candidate pairs generated by Rules treated as successful interactions to define user similarity
ProCF+	Hybrid method combining Rules and ProCF, using a probabilistic user similarity function based on successful and unsuccessful interactions to optimize success rate improvement

Table 1. Methods Evaluated in Online Trial.

Compatible Subgroup Rules (Rules)

This recommender works by dynamically constructing rules for each user of the form: *if* $u_1, \ldots, u_n$ (condition) then $c_1, \ldots, c_n$ (conclusion), where the u_i are profile features of the user and c_i are corresponding profile features of the candidate (Kim et al. 2012b). If the user satisfies the condition of such a rule, any candidate satisfying the conclusion can be recommended to the user. Candidates are ranked based on match specificity and their positive reply rate.

Each profile feature is an attribute with a specific value, for example, age = 30–34, location = Sydney (each attribute with a discrete set of possible values). An initial statistical analysis determines, for each possible attribute *a* and each of its values *v* (taken as a sender feature), the best matching values for the same attribute *a* (taken as receiver features), here treating male and female sender subgroups separately. For example, for males the best matching values for senders with feature age = 30–34 might be females with age = 25–29.

This method can recommend and provide candidates to a wide range of users, as, in contrast to the pure form of CF used in the first trial, users and candidates are not restricted to have prior interactions. The drawbacks are the high computational cost of computing subgroup rules and the lower success rate improvement.

Profile-Based User Similarity CF (SIM-CF)

In contrast to the CF recommender used in the first trial, where similarity of users was defined using their common positive contacts (Krzywicki et al. 2010), similarity of users in SIM-CF is defined using a very simple profile-based measure based only on age difference and location (Kim et al. 2012a).

Definition 1

For a given user *u,* the class of similar users consists of those users of the same gender and sexuality who are either in the same 5-year age band as *u* or one age band either side of *u,* and who have the same location as *u.*

Age and location are used as the basis of similarity since these two attributes are the most commonly used in searches on the site. The data shows that successful interactions are far more likely between people with at most a 10-year age difference than between those with a greater age difference. Similarly, location is not arbitrary but designed to capture regions of similar socioeconomic status. Thus there is reason to believe that users similar under this measure will encompass a large number of users with similar behavior and socioeconomic status. Note that we investigated a number of more sophisticated definitions of user similarity, but surprisingly, none of them gave any improvement in the context of SIM-CF so were not pursued further (however, see the discussion of ProCF+ below, where both successful and unsuccessful interactions are used to define user similarity).

SIM-CF is standard user-based CF (figure 2) and works by finding users similar to a given user and recommending the contacts of those users. Candidates are first rated by the number of their successful interactions that are initiated by users similar to the target user, then reranked using the decision tree critic, which demotes candidates with a high likelihood of an unsuccessful interaction. Historical data analysis showed that this would give a higher success rate improvement and diversity of candidates (Krzywicki et al. 2012).

Rule-Based Similar Recipients CF (RBSR-CF)

RBSR-CF calculates similarity for a user *u* based on the successful interactions of other users with candidates generated by the Rules recommender for *u,* then applies CF to generate and rank candidates (Kim et al. 2012a), as in a content-boosted method (Melville, Mooney, and Nagarajan 2002). In figure 2, Rules recommendations are shown as a double arrow labeled R. As in SIM-CF, candidates are ranked using the number of successful interactions with users similar to the target user, and the decision tree rules are used for reranking. A strength of this method is that it provides a greater diversity of candidates than SIM-CF with a similar success rate improvement, but with the drawback of a higher computational complexity and lower usage of recommendations.

Probabilistic CF+ (ProCF+)

ProCF (Cai et al. 2013) uses a more sophisticated model of user similarity than SIM-CF, derived from successful and unsuccessful interactions (denoted with arrows labeled + and – in figure 2). As with RBSR-CF, ProCF+ uses actual interactions augmented with user-candidate pairs generated by the Rules recommender treated as successful interactions, applying CF to generate and rank candidates (as with SIM-CF and RBSR-CF but without the decision tree critic). The main advantage of ProCF+ is the higher success rate improvement than SIM-CF and RBSR-CF (due to the more accurate calculation of user similarity), but this comes with a higher computational cost and lower usage of recommendations. In addition, ProCF+ generates more user-candidate pairs further apart in geographical distance: the data suggests that these particular matches are likely to be successful, despite the fact that long-distance matches generally are unlikely to be successful.

Baseline Method

In addition to our four methods, a number of profile-based proprietary methods were trialled, built around matching heuristics and individual contact preferences. One method based on profile matching was agreed as a baseline for comparison with our algorithms. But, as recommendations for this method were not able to be recorded, comparison of our methods to the baseline covers only contacts and open communications.

Selection of Recommender for Deployment

A live trial of recommenders was conducted as a close collaboration between researchers and the dating site company, and treated by the company as a commercial project with strictly defined and documented objectives, requirements, resources, methodology, key performance indicators and metrics all agreed in advance. The main objective of the trial was to determine if a novel recommender could perform better than the baseline method, and if so, to select one such recommender for deployment. Aside from an increase in revenue, the company was aiming to improve overall user experience on the site, and to respond to competitor site offerings of similar functionality.

Considerable time and effort of the company was dedicated to the proper conduct of the trial, including project management, special software development and additional computational resources. A whole new environment was created including a separate database containing generated recommendations, impressions and clicks for all methods, running alongside the production system so as to minimally impact system performance.

Trial Methodology

Each of the methods described in the Recommendation Methods section received 10 percent of all site users, including existing and new users joining in the period of the trial. To avoid cross-contamination of user groups, once a user was assigned to a group, they remained in the same group for the duration of the trial. Thus the proportion of new users in each group increased over the course of the trial. The recommenders were required to compute recommendations daily, and hence provide recommendations to new users with very limited training data.

After a brief period of onsite testing and tuning, the trial was conducted over 6 weeks, from May to mid-June 2012. In contrast to the first trial, recommendations were allowed to be repeated from day to day with the restriction not to generate candidates with whom the user had had a prior interaction. Our recommenders generated candidates on the day they were delivered, using an offline copy of the database created that morning; thus training data was one day out of date. In contrast, the baseline method generated and delivered recommendations on the fly. In consequence, the baseline method could recommend users who had joined the site after our recommenders had been run and make use of data unavailable to our recommenders, giving it some advantage. The number of recommendations generated was limited to the top 50 candidates for each user.

Candidates for each user were assigned a score, which, for our recommenders, was the predicted likelihood of the interaction being successful. Candidates were displayed on a number of user pages, four at a time, with probability proportional to their score, and users could see more candidates by clicking on an arrow button on the interface.

Trial Metrics

A set of primary metrics were agreed between the research group and the company before the trial in a series of meetings. There are two types of primary metric, corresponding to the results presented in tables 2 and 3. The first set of metrics are used for comparing the user groups allocated the recommendation methods to the baseline user group, and focus on usage of recommendations. The second set of metrics are used to compare user behavior within the same group, focusing on success rate improvement over search and usage of recommendations, measured as the proportion of contacts or open communications initiated from recommendations.

A third group of metrics (table 4) are additional metrics, determined only after the trial to be of relevance in assessing the increase in user engagement with the site due to the recommenders. Note that it is not that these metrics could not have been defined before the trial, rather their usefulness only became evident after the trial. These metrics cover contacts and open communications to the majority of non-

	Rules	SIM-CF	RBSR-CF	ProCF+
Lift in contacts per user	3.3%	10.9%	8.4%	–0.2%
Lift in positive contacts per user	3.1%	16.2	10.4%	5.6%
Lift in open communications per user	4.3%	4.8%	3.7%	0.8%

Table 2. Comparison of Recommender Groups with Baseline Group.

	Rules	SIM-CF	RBSR-CF	ProCF+
Success rate improvement over search	11.2%	94.6%	93.1%	133.5%
Contacts from recommendations	8.1%	11.8%	9.9%	8.2%
Positive contacts from recommendations	8.9%	20.7%	17.5%	17.2%
Open communications from recommendations	8.1%	18.2%	14.8%	13.4%

Table 3. Comparisons Within Groups.

	Rules	SIM-CF	RBSR-CF	ProCF+
Contacts with no reply	33.0%	26.1%	27.1%	27.3%
Positive contacts to nonpopular users	85.7%	62.0%	64.4%	63.9%
Positive contacts by women	33.1%	33.4%	30.0%	27.0%
Recommendations with age difference > 10 years	0.1%	3.3%	3.2%	8.3%
Average/median distance in km in recommendations	91/20	106/20	384/40	478/50

Table 4. Additional Metrics.

highly popular users (influencing overall user engagement), contacts and open communications initiated by women (related to maintaining the pool of women on the site), and age/location differences between users and candidates (since some users reacted strongly when receiving candidates very different in age or location from their stated preferences).

Trial Results and Selection of Best Method

Data from the trial for final evaluation of the recommendation methods was collected two weeks after the end of the trial to count responses to messages initiated during the trial and to count open communications resulting from recommendations delivered during the trial. Note that due to the considerable variation in outlier behavior (a small number of highly active members), the top 200 most active users from the whole trial were removed from the analysis.

Table 2 summarizes the results for comparison of the recommender groups with the baseline group. These metrics are percentage lifts in overall behavior, and reflect how much users act on recommendations. Here SIM-CF produced the best results. The increase in open communications is important, because this is directly related to revenue. Even a small increase in this measure was considered significant from the business perspective.

The second set of metrics for comparing user behavior within groups (table 3) include success rate improvement and usage of recommendations, for which these measure the proportion of the behavior of users in the same group produced by recommendations. Again SIM-CF performed the best, while ProCF+ showed the best success rate improvement, consistent with historical data analysis. What is most surprising is the higher than expected usage of recommendations for Rules and the lower than expected performance of ProCF+ on these metrics (that is, expected from historical data analysis), suggesting that ProCF+ is overly optimized to success rate improvement. Also interesting is that, while ProCF+

users make heavy use of the recommendations, their overall increase in behavior is much less, suggesting some cannibalization of search behavior by the recommender, whereas in the other groups, the recommenders result in more additional user behavior. The potential for cannibalization of search by the recommenders was a major concern to the dating site company, because if this happened, overall revenue would not increase.

The additional metrics (table 4) relate more to long-term user experience, user satisfaction with the recommendations, and maintaining overall user engagement. It is understood that the metrics do not capture these qualities directly, and that the interpretation of the results is more subjective.

The first metric is the proportion of contacts from recommendations without any reply; the next relates to contacts to nonpopular users. The importance of these metrics is that many contacts, typically those to popular users, go without a reply, potentially discouraging users. It was felt that even a negative reply would make the user more engaged with the site. On these metrics, all of our CF methods perform very well, as all are designed not to overrecommend popular users (thus avoiding contacts with no response). For SIM-CF and RBSR-CF, this is due to the use of the decision tree rules that demote popular users in the rankings; for ProCF+ due to the use of unsuccessful interactions in the calculation of user similarity. Though SIM-CF is best on the proportion of contacts with no reply, the other CF methods are ahead on contacts to nonpopular users. This may suggest that when popular users are recommended by SIM-CF, they are slightly more likely to generate a reply.

The third metric concerns usage of recommendations by women. Women are often thought of as being passive on online dating sites, however this is not the case. Women are more selective in their contacts and are thus typically less active than men. SIM-CF is clearly the best method for encouraging actions initiated by women.

The remaining metrics relate to general user perception and trust in the reliability of the recommender system, as candidates shown that were outside a user's stated preferences were sometimes regarded by users as faults in the system (these candidates were generated because data indicates that many of them will be successful). Some simple measures of age and location differences were calculated for the whole set of recommendations generated. Rules is the best method on these metrics, while of the CF methods, RBSR-CF is superior on age difference and SIM-CF on location difference. ProCF+ has the highest proportion of recommendations with an age difference more than 10 years, which, since it also has the highest success rate lift, may suggest that these recommendations have a high success rate.

On the basis of this evaluation, SIM-CF was selected as the method for deployment. It has the best score on all primary metrics except success rate improvement, the smallest proportion of contacts with no reply, and the best proportion of contacts initiated by women. This method also gives a balanced approach to age and location differences due to how user similarity is calculated. The values for the other metrics were lower than for other methods, but not deemed to be substantially lower. Also of importance was the fact that SIM-CF was the least complex CF method to implement, with no dependencies on other processes, whereas RBSR-CF and ProCF+ both depend on Rules.

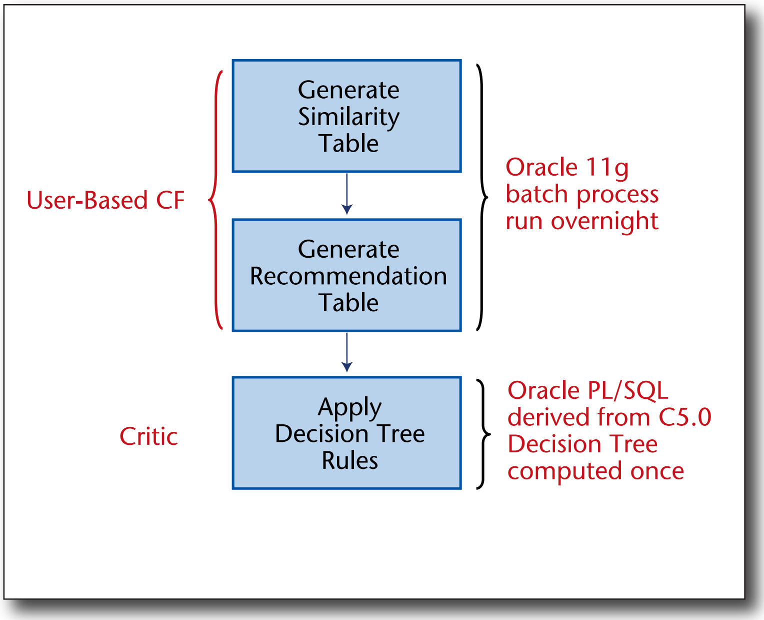
User-Based CF
Generate Similarity Table
Generate Recommendation Table
Oracle 11g batch process run overnight
Critic
Apply Decision Tree Rules
Oracle PL/SQL derived from C5.0 Decision Tree computed once

Figure 3. SIM-CF Trial Implementation.

Recommender Deployment

The initial SIM-CF implementation shown in figure 3 was used for evaluation on historical data and in the trial. SIM-CF generates ranked recommendations in a two stage process. The first stage involves generating candidates with a preliminary score using profile-based user similarity; the second stage involves using decision tree rules computed from a larger training set to weight the scores produced in the first stage (Krzywicki et al. 2012). Note that the decision tree rules are the same on each run of the recommender, since retraining the decision tree is done only as needed. Hence this step of the process is comparatively simple.

SIM-CF used an Oracle 11g database to store tables for the user similarity relation and for recommendations. In the trial context, each table had several tens of millions of rows, well within performance requirements. The reason for using Oracle was that this is

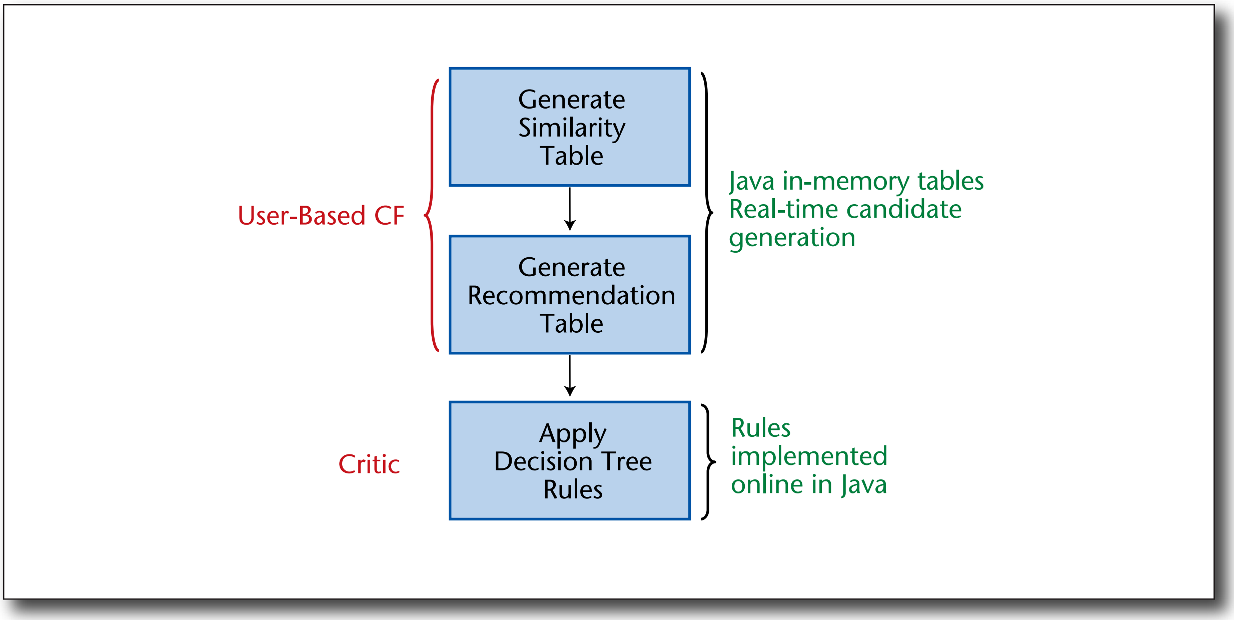
Generate Similarity Table
Generate Recommendation Table
Apply Decision Tree Rules
User-Based CF
Critic
Java in-memory tables Real-time candidate generation
Rules implemented online in Java

Figure 4. SIM-CF Deployment Implementation.

the database system used by the dating site company. This implementation also enabled us to experiment extensively with variations of the different methods. Our implementation was efficient and robust enough to be used in the trial and in the initial deployment environment. The decision tree was constructed using C5.0 from RuleQuest Research, with rules derived from the decision tree converted into Oracle PL/SQL.

Deployment Process

The initial implementation of SIM-CF was high-quality, robust software, suitable for research and development and a rigorous trial for 10 percent of the users, but it was not suited to the production environment due to its heavy reliance on the database. The company decided that the method could be implemented in Java with the use of in-memory tables to provide near-real-time recommendations on a per user basis as needed. Here *near real time* means that recommendations are computed and delivered to users after they take an action to move to a new page that contains recommendations, with no appreciable delay in page loading time. This required a reimplementation of SIM-CF and integration with the production system (figure 4). This had to be done with minimal cost and impact on the existing backend system, which served millions of online customers.

Some changes were made to simplify the SIM-CF user similarity function based on age bands and location by calculating the exact age difference and estimating the distance between users. This allowed SIM-CF to provide more candidates, since, if the number of similar users based on the same location as the user was insufficient, users further away could be used to generate candidates. Another change was to augment successful interactions with open communications for generation of candidates, after some experiments confirmed that this would slightly improve the results.

The whole development and deployment process took around 3.5 months and was done using incremental releases and testing. The actual design and development, including decisions about changes, was done by the dating site company. The role of the research group was to advise on the changes and provide detailed information about the SIM-CF method. The following describes the deployment timeline:

Mid-June 2012: Trial ended and analysis and selection of recommender commenced.

August 2012: SIM-CF was selected and was switched to deliver recommendations to all users from the offline database, except for the first day of a user's registration when recommendations were supplied by the baseline method. At the same time, design and development of the in-memory version of SIM-CF started.

September 2012: The first version of in-memory SIM-CF started to deliver recommendations to 60 percent of users, working in parallel with the original trial version for the next couple of weeks and still using the offline database.

October 2012: The in-memory version was switched to 100 percent of users, still running from the offline database.

	Trial	Deployment
Comparison with Baseline Group		
Lift in contacts per user	10.9%	10.3%
Lift in positive contacts per user	16.2%	12.4%
Lift in open communications per user	4.8%	7.3%
Comparisons Within Groups		
Success rate improvement over search	94.6%	88.8%
Contacts from recommendations	11.8%	18.6%
Positive contacts from recommendations	20.7%	30.2%
Open communications from recommendations	18.2%	28.3%
Additional Metrics		
Contacts with no reply	26.1%	24.9%
Positive contacts to nonpopular users	62.0%	63.5%
Positive contacts by women	33.4%	39.3%

Table 5. Comparison of Trial and Deployment Metrics.

November 2012: The production version of SIM-CF started to run from the live database. As the recommender started to use the online database, recommendations could be generated more often, covering newly joined users and eliminating the need for recommendations for new users generated by the baseline method. An initial concern was that memory usage might be too high, however a careful design ensured that the memory requirement was within reasonable limits.

The dating site company indicated that the recommender system is stable and does not require any immediate additional maintenance related to the method itself. The decision tree rules have been tested against several data sets from different periods and given consistent results. Therefore there is currently no provision to determine when the decision tree rules need to be updated. If such a need occurs in the future, it would not be difficult to update the rules.

Postdeployment Evaluation

In this section, we compare the results from the trial and posttrial deployment to show how the benefits of the SIM-CF recommender established during the trial are maintained in the deployed setting. Our comparison covers the key metrics discussed in Trial Results and Selection of Best Method section and is based on data from three months collected between November 2012 (after the production version started using the live database) and February 2013. As in the trial analysis, we allowed 2 extra weeks for collecting responses to messages and open communications to candidates recommended during the three months.

Table 5 compares posttrial deployment metrics to those shown previously for the trial (repeated in the first column), except there is one important difference. In the trial setting, the first group of primary metrics compared the recommender group to the baseline group. Now since all users have SIM-CF there is no baseline group.

Therefore we calculate the lift in various measures for the deployment setting with respect to data from November 2011 to February 2012, when the baseline recommender was in use. The reason for using this period of time is that there is no need to adjust for seasonal effects (typically the values of such metrics vary throughout the year). Though inexact, this gives us reasonably high confidence that the group metric results are maintained after deployment.

The next set of primary metrics concerning usage of recommendations shows a drop in success rate improvement in the posttrial deployment setting but an increase in usage of recommendations. The exact reasons for these changes are unknown, but could be due to the modifications to the original SIM-CF method (see the Recommender Deployment section), which were made with a view to increasing contacts at the cost of slightly lowering success rate improvement.

The final section of table 5 compares the trial and posttrial deployment values for the additional metrics. The values of all metrics improved since the trial. One reason for this could be that recommendations are generated from an online database (as opposed to the offline database used during the trial), thus covering new users soon after they join the site. Providing recommendations to new users at this time is very important as they are less experienced in their own searches and eager to make contacts.

Lessons Learned

Looking over the whole period of this project from inception to deployment, we identify several major lessons learned during the process of the development and deployment of an AI application in a commercial environment that we believe to be general but also more relevant to the field of recommender systems. These lessons can be summarized as: (1) the results of evaluation on historical data do not necessarily translate directly to the setting of the deployed system, since deployment of the system changes user behavior, (2) commercial considerations go far beyond simple metrics used in the research literature, such as precision, recall, mean absolute error or root mean squared error, (3) computational requirements in the deployment environment, especially scalability and runtime performance, determine what methods are feasible for research (in our case, collaborative filtering methods that are popular with researchers, such as types of matrix factorization, were infeasible for the scale of our problem in the deployed setting).

Historical Data Analysis Insufficient

The fundamental problem with evaluation of a recommendation method using historical data is that what is being measured is the ability of the method to predict user behavior without the benefit of the recommender (in our case, behavior based on search). There is no a priori guarantee that such results translate to the setting of deployment, where the objective is to change user behavior using the recommender. Critical was the first trial (Krzywicki et al. 2012) where we learned that, though the values of our metrics from the trial were not the same as those on historical data, overall trends were consistent, meaning that evaluation on historical data was a reliable indicator of future recommender performance.

After we had developed our best methods, the trial reported in this article was essential for selecting the method for deployment, due to the impossibility of choosing between the methods using historical data analysis alone.

Another facet of the problem is that typically evaluations on historical data consider only static data sets. The highly dynamic nature of the deployed system is ignored, in particular the high degree of change in the user pool as users join or leave the site, and the requirement for the recommender to generate candidates over a period of time as users change their overall interaction with the system. Both trials showed that our methods were capable of consistent performance over a extended period of time with a highly dynamic user base.

Overly Simplistic Metrics

Concerning metrics, our basic observation is that the research literature over-emphasizes simple metrics that fail to capture the complexity of the deployment environment. Simple metrics are usually statistical measures that aggregate over a whole user base, so do not adequately account for the considerable variation between individual users. In our case, some measures can be dominated by a minority of highly active users. However a deployed system has to work well for all users, including inactive users for whom there is little training data. Moreover, often these metrics are used to aggregate over all potential recommendations, however what matters are only the recommendations the user is ever likely to see (the top *N* candidates), not how well the method predicts the score of lower ranked candidates.

We found particularly useful a prototype that we developed to enable us to visually inspect the recommendations that would be given to an individual user, to see if those recommendations might be acceptable. In this way, we identified very early the problem of relying only on the simple metric of success rate improvement, which tended to result in recommendations that were all very similar and which may not have been of interest to the user. Thus even considering simple metrics, what was needed was a way of taking into account several metrics simultaneously, involving design trade-offs in the recommendation methods.

Further, the company deploying the recommender was of course interested in short-term revenue, but also in improving the overall user experience which, it was understood, would lead to increased engagement with the site and potentially more revenue in the long term. However, the simple metrics used in the literature can be considered only proxies indirectly related even to short-term revenue, so much interpretation and discussion was needed to understand the impact of the recommenders on user experience (which motivated the additional metrics described above). The company chose the method for deployment by considering a range of metrics covering both short-term revenue and user experience.

Commercial Feasibility

Our final point is that there is often a large gap between typical research methodology and commercial requirements. Our project was successful because we took seriously the requirements of the deployment environment and focused research on those methods that would be feasible to trial and deploy. The alternative approach of developing a research prototype (without considering feasibility), then treating the transfer of that prototype to an industrial context as merely a matter of implementation, would not have worked.

Even so, the research environment has different requirements from the deployment environment, which means that some reimplementation of the research system is almost inevitable for deployment. The research system is focused on experimentation and requires simple, flexible, and easily modifiable software, whereas the emphasis in deployment is on resource constraints and online efficiency. Though our implementation worked in the trial and in a deployed setting where recommendations were up to one day out of date, our implementation would not work in the production environment, and moreover, we could not have built a system in the research laboratory that would work in production since this required integration with the commercial systems.

Conclusion

We have presented the results of a successful deployment of our people-to-people recommender system on a large commercial online dating site with nearly half a million active users sending more than 70,000 messages a day. The recommender had been in use for about 7 months (from August 2012 to March 2013) before these results were obtained. In the period from November 2012 to March 2013, 61 percent of active users clicked recommendations and 33 percent of them communicated with recommended candidates.

The recommender system is a hybrid system combining several AI techniques that all contributed to its overall success. First, collaborative filtering allows recommendations to be based on user behavior rather than profile and expressed preferences. Second, decision tree rules were crucial in addressing the common problem with collaborative filtering in over-recommending popular items, which is particularly acute for people-to-people recommendation. No single AI method, whether decision tree learning, profile-based matching, or collaborative filtering, could alone produce satisfactory results. More generally, we think there is much scope for further research into hybrid recommender systems that combine multiple sources of information when generating and ranking recommendations.

Methods developed for this recommender system can be used, apart from in online dating, in other social network contexts and in other reciprocal recommendation settings where there are two-way interactions between entities (people or organizations) with their own preferences. Typical such problems include intern placement and job recommendation. Moreover, our method of using decision tree rules as a critic to reduce the recommendation frequency of popular users can also be applied to item recommendation.

Acknowledgements

This work was funded by Smart Services Cooperative Research Centre. We would also like to thank our industry partner for supporting this research consistently throughout the whole process, and specifically for conducting the trial and giving us details of the method reimplementation and the deployment process.

References

Burke, R. 2002. Hybrid Recommender Systems: Survey and Experiments. *User Modeling and User-Adapted Interaction* 12(4): 331–370.

Cai, X.; Bain, M.; Krzywicki, A.; Wobcke, W.; Kim, Y. S.; Compton, P.; and Mahidadia, A. 2010. Collaborative Filtering for People to People Recommendation in Social Networks. In *AI 2010: Advances in Artificial Intelligence,* ed. J. Li, 476–485. Berlin: Springer-Verlag.

Cai, X.; Bain, M.; Krzywicki, A.; Wobcke, W.; Kim, Y. S.; Compton, P.; and Mahidadia, A. 2013. ProCF: Generalising Probabilistic Collaborative Filtering for Reciprocal Recommendation. In *Advances in Knowledge Discovery and Data Mining,* ed. J. Pei, V. Tseng, L. Cao, H. Motoda, and G. Xu. Berlin: Springer-Verlag.

Finkel, E. J.; Eastwick, P. W.; Karney, B. R.; Reis, H. T.; and Sprecher, S. 2012. Online Dating: A Critical Analysis from the Perspective of Psychological Science. *Psychological Science in the Public Interest* 13(1): 3–66.

Kim, Y. S.; Mahidadia, A.; Compton, P.; Cai, X.; Bain, M.; Krzywicki, A.; and Wobcke, W. 2010. People Recommendation Based on Aggregated Bidirectional Intentions in Social Network Site. In *Knowledge Management and Acquisition for Smart Systems and Services,* ed. B.-H. Kang and D. Richards, 247–260. Berlin: Springer-Verlag.

Kim, Y. S.; Krzywicki, A.; Wobcke, W.; Mahidadia, A.; Compton, P.; Cai, X.; and Bain, M. 2012a. Hybrid Techniques to Address Cold Start Problems for People to People Recommendation in Social Networks. In *PRICAI 2012: Trends in Artificial Intelligence,* ed. P. Anthony, M. Ishizuka, and D. Lukose, 206–217. Berlin: Springer-Verlag.

Kim, Y. S.; Mahidadia, A.; Compton, P.; Krzywicki, A.; Wobcke, W.; Cai, X.; and Bain, M. 2012b. People-To-People Recommendation Using Multiple Compatible Subgroups. In *AI 2012: Advances in Artificial Intelligence,* ed. M. Thielscher and D. Zhang, 61–72. Berlin: Springer-Verlag.

Krzywicki, A.; Wobcke, W.; Cai, X.; Bain, M.; Mahidadia, A.; Compton, P.; and Kim, Y. S. 2012. Using a Critic to Promote Less Popular Candidates in a People-to-People Recommender System. In *Proceedings of the Twenty-Fourth Annual Conference on Innovative Applications of Artificial Intelligence,* 2305–2310. Palo Alto, CA: AAAI Press.

Krzywicki, A.; Wobcke, W.; Cai, X.; Mahidadia, A.; Bain, M.; Compton, P.; and Kim, Y. S. 2010. Interaction-Based Collaborative Filtering Methods for Recommendation in Online Dating. In *Web Information Systems Engineering,* ed. L. Chen, P. Triantafillou, and T. Suel, 342–356. Berlin: Springer-Verlag.

Melville, P.; Mooney, R. J.; and Nagarajan, R. 2002. Content-Boosted Collaborative Filtering for Improved Recommendations. In *Proceedings of the Eighteenth National Conference on Artificial Intelligence* (AAAI-02), 187–192. Menlo Park, CA: AAAI Press.

Oyer, P. 2014. *Everything I Ever Needed to Know About Economics I Learned from Online Dating*. Boston: Harvard Business Review Press.

Pizzato, L.; Rej, T.; Akehurst, J.; Koprinska, I.; Yacef, K.; and Kay, J. 2013. Recommending People to People: the Nature of Reciprocal Recommenders with a Case Study in Online Dating. *User Modeling and User-Adapted Interaction* 23(5): 447–488.

Webb, A. 2013. *Data, A Love Story.* New York: Plume, Penguin Group.

Wayne Wobcke is an associate professor in the School of Computer Science and Engineering at the University of New South Wales, Australia. His research interests span intelligent agent theory, dialogue management, personal assistants, and recommender systems. From 2008 to 2014, he led the Personalisation program within Smart Services Cooperative Research Centre, which provided the environment for this collaboration. He has a Ph.D. in computer science from the University of Essex.

Alfred Krzywicki is a research fellow in the School of Computer Science and Engineering at the University of New South Wales, Australia, with a background in both software engineering and AI. After receiving his Ph.D. from the University of New South Wales in 2012, he worked on social media analysis and contributed to the implementation and commercial deployment of a recommender system. His current research interests include recommender systems, text, and stream mining.

Yang Sok Kim is an assistant professor at the School of Management Information Systems at Keimyung University, Korea. His research interests include knowledge-based systems and machine learning and their hybrid approaches. Kim received his Ph.D. in computing in 2009 from the School of Computing and Information Systems at the University of Tasmania. He worked as a postdoctoral researcher for the Personalisation program within Smart Services Cooperative Research Centre.

Xiongcai Cai is a scientist with a general background in AI with expertise in the fields of machine learning, data mining, and social media analysis. He received a Ph.D. in computer science and engineering from the University of New South Wales, Australia, in 2008. He has spent the past decade researching and developing models to better understand behaviors and patterns that expand the scope of human possibilities.

Michael Bain is a senior lecturer in the School of Computer Science and Engineering at the University of New South Wales, Australia. His research focuses on machine learning, particularly the logical foundations of induction, and he has experience in a wide variety of application areas including computer chess, bioinformatics, and recommender systems. He has a Ph.D. in statistics and modeling science from the University of Strathclyde.

Paul Compton is an emeritus professor in the School of Computer Science and Engineering at the University of New South Wales, Australia. Most of his research has been around the idea of building systems incrementally using a learning or knowledge acquisition strategy known as Ripple-Down Rules.

Ashesh Mahidadia is the chief executive officer of the company smartAcademic, which empowers businesses to add intelligence to the way they operate. He has a Ph.D. in artificial intelligence from the University of New South Wales, Australia. His research interests include knowledge-intensive machine learning, recommender systems, intelligent assistants, and intelligent education systems. He was a senior researcher in the Personalisation program within Smart Services Cooperative Research Centre.