Evaluation and Deployment of a People-to-People Recommender in Online Dating

A. Krzywicki, W. Wobcke, Y. S. Kim*, X. Cai, M. Bain, P. Compton and **A. Mahidadia**
School of Computer Science and Engineering
University of New South Wales
Sydney NSW 2052, Australia
{alfredk,wobcke,xcai,mike,compton,ashesh}@cse.unsw.edu.au
*School of Computing and Information Systems
University of Tasmania
Hobart TAS 7001, Australia
YangSok.Kim@utas.edu.au

Abstract

This paper reports on the successful deployment of a people-to-people recommender system in 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 key metrics generally hold their value or show an increase compared to the trial results, and that the recommender system delivered its projected benefits.

1 Introduction

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. This paper, however, concerns people-to-people recommendation in an online dating context. The main difference between the two types of recommender is that people-to-people recommenders are reciprocal (Pizzato et al. 2013): users are involved in two-way interactions and their invitations may be accepted or rejected. Another difference is that, since users can only maintain contact with a small number of people, candidates should not be recommended too often, whereas the same item can be recommended any number of times.

A typical online dating system consists of a backend, where user profiles, contacts and transactions are stored in a database, and a frontend, with a web-based user interface and tools allowing users to specify their preferences, perform searches and contact other users. Our recommender is part of the backend, providing a number of ranked candidates to each user, that are shown on various web pages.

The rest of the paper is structured as follows. In the next section, we outline the basic problems of recommendation in online dating. Next we present our recommendation methods, then discuss a live trial conducted with the aim of selecting one method for deployment on the site (Section 4). Section 5 contains details of the deployment process for the "winning algorithm". A post-deployment evaluation of the method is provided in Section 6, followed by lessons learned, a brief discussion of other deployed recommenders in social networking, and the conclusion.

2 Problem Description

We first briefly describe the user experience on the online dating site we are considering. Users typically begin with simple registration, which allows them to communicate with other users. During registration, a user enters basic details, such as name, date of birth and address. Other details, such as partner preferences, are optional and can be entered any time later. Then users typically view the profiles of other users, those either promoted by the site (the most popular and new users in each age group) or resulting from search. At this stage, a user can send predefined messages free of charge (we call them contacts) to selected candidates, or proceed directly to paid open communication with another user. A typical contact message would be "I would like to get to know you, would you be interested?", which may receive a response such as "Yes, I am waiting for your e-mail" (a positive response), or "I am not interested, but wish you good luck in your search" (a negative response), or which may receive no response at all. Whether a response is positive or negative is predetermined by the dating site, but importantly, we count a contact that receives no response as negative. This data allows us to accurately measure the success rate of contacts, defined as the proportion of contacts with positive responses out of all contacts. Open communication (site mediated e-mail) allows the user to exchange contact details with the other person, after which they can communicate offline.

Typically users find potential contacts using keyword search, however searches usually return a large number of candidates, presenting the user with the problem of choice. Users may become frustrated by spending time and not finding desirable partners, or by being rejected or ignored by those people they do contact. This may lead to an increased attrition rate of users on the site, which in turn reduces the candidate pool for others. Therefore a major goal of a recommender system is to improve a user's chances of a successful interaction, and a secondary objective is to increase overall user engagement with the site. A particular problem in such people-to-people recommendation is that users tend to prefer (and hence contact) a small number of "highly popular" users, however these contacts are likely to be rejected or ignored. Thus over-recommendation of highly popular users, a common property of collaborative filtering recommendation methods, should be avoided.

3 Development of Recommenders

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 for the majority of (non-highly popular) users. In this section, we discuss the design tradeoffs that led to the development of the methods, and outline their properties.

3.1 Design Considerations

The methods we present in this section 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 (Kim et al. 2010; Krzywicki et al. 2010). Scalability of the method is a non-negotiable requirement: in a typical setting, the recommender system needs to cover hundreds of thousands of active users and process several millions of user contacts. 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.

A potential worry with our approach was that all preliminary evaluation of methods was done on historical data provided by the dating site company, however there was no guarantee that such evaluation would transfer to the setting of a deployed recommender. This is because the evaluation on historical data was essentially on a prediction task, with the object being to predict the successful interactions that users found by search, that is, in the absence of a recommender. We conducted a first trial of two methods in 2011 over a 9 week period, where recommendations were delivered via e-mail (Krzywicki et al. 2012). The important results were that: (i) the performance of the methods in the trial setting was consistent with that on historical data, giving us confidence in our methodology, and (ii) both methods were able to provide recommendations of consistent quality over a period of time.

The CF method trialled did not address the "cold start" problem (recommendation to and of new users), so in subsequent research, we developed and tested numerous hybrid CF methods that could recommend to almost all users while maintaining a high success rate (Kim et al. 2012a). This paper 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. Our methods were developed considering numerous dimensions, often conflicting, which we knew to influence the quality of a recommendation method, represented by the following metrics: (i) success rate improvement (how much more likely the user is to have a successful interaction when following recommendations), (ii) recall (the degree to which the user likes the recommended candidates), (iii) user coverage (the proportion of users who can receive recommendations), and (iv) diversity (the number of distinct candidates and their distribution of their recommendations). Each of our methods makes different design tradeoffs between these dimensions.

3.2 Summary of Methods

Our four trialled methods are summarized in Table 1 for reference, and are discussed further below.

Table 1: Methods evaluated in the trial

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 method using recommendations of Rules to "boost" CF
ProCF+	Hybrid combining Rules with ProCF, which is based on a probabilistic user similarity function to optimize success rate improvement

Compatible Subgroup Rules (Rules): This recommender works by dynamically constructing rules for each user of the form: *if* $u_1, \cdots, u_n$ (condition) *then* $c_1, \cdots, c_n$ (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, e.g. 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 recommendations to a wide range of users, as the candidates are not restricted to those with 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 (Krzywicki et al. 2010), the similarity of users formerly determined by positive contacts in common is replaced in SIM-CF by a profile-based user similarity measure (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 socio-economic status.

SIM-CF 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. Candidates are then re-ranked 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. Since highly popular candidates often result in unsuccessful interactions, one effect of the re-ranking is to "demote" highly popular candidates, so that they are not over-recommended. Mutiplying scores by rule weights is justified by Bayesian reasoning (Krzywicki et al. 2012).

Rule-Based Similar Recipients CF (RBSR-CF): RBSR-CF exploits user-candidate pairs generated by the Rules recommender as if they were real interactions, in conjunction with CF (Kim et al. 2012a), to provide a "content-boosted" method as defined in (Melville, Mooney, and Nagarajan 2002). 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 re-ranking. 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 recall.

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. As with RBSR-CF, ProCF+ makes use of user-candidate pairs generated by the Rules recommender as if they were real interactions to calculate this similarity measure, then applies CF to generate and rank candidates (as with SIM-CF and RBSR-CF but without the Decision Tree rules). 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 recall. In addition, ProCF+ generates more user-candidate pairs further apart in geographical distance, leading to a lower recall (even though the data suggests these matches are likely to be successful).

The 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.

4 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.

4.1 Trial Methodology

Each of the methods described in Section 3 received 10% 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 some advantage to the baseline method. 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.

4.2 Trial Metrics

A set of primary metrics were agreed between the research group and the company before the trial in a series of meetings. These metrics are shown in Table 2 and are divided into two groups: group comparison metrics comparing lift in various measures for one group compared to the baseline (first section of the table), and single group metrics concerning usage of the recommendations (second section of the table).

Additional metrics (third section of the table) focusing on user experience were determined after the trial to measure specific aspects of the methods, such as contacts and communications to non-highly popular users (a measure of likely

Table 2: Comparison of methods based on trial results

	Rules	SIM-CF	RBSR-CF	ProCF+
Primary metrics: Comparison of recommender groups with baseline group				
Lift in contacts initiated per user	3.3%	**10.9%**	8.4%	-0.2%
Lift in positive contacts initiated per user	3.1%	**16.2%**	10.4%	5.6%
Lift in open communications initiated per user	4.3%	**4.8%**	3.7%	0.8%
Primary metrics: Usage of recommendations				
Lift in success rate over users' own search	11.2%	94.6%	93.1%	**133.5%**
Proportion of contacts initiated from recommendations	8.1%	**11.8%**	9.9%	8.2%
Proportion of positive contacts initiated from recommendations	8.9%	**20.7%**	17.5%	17.2%
Proportion of open communications initiated from recommendations	8.1%	**18.2%**	14.8%	13.4%
Additional metrics: Usage of recommendations				
Proportion of messages with no reply	33.0%	**26.1%**	27.1%	27.3%
Proportion of contacts initiated to non-highly popular users	**78.7%**	57.2%	62.5%	65.8%
Proportion of positive contacts initiated to non-highly popular users	**85.7%**	62.0%	64.4%	63.9%
Proportion of open communications initiated to non-highly popular users	**85.6%**	61.8%	61.7%	63.5%
Proportion of contacts initiated by women	25.3%	**26.6%**	23.1%	24.7%
Proportion of positive contacts initiated by women	33.1%	**33.4%**	30.0%	27.0%
Proportion of open communications initiated by women	23.6%	**28.2%**	24.0%	22.8%
Average age difference in recommendations	**2.65**	3.9	3.6	4.6
Proportion of 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

increased overall user engagement), contacts and 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).

4.3 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.

Table 2 summarizes the results of the trial on a wide range of metrics. Note that due to the considerable variation in outlier behaviour (a small number of highly active members), the top 200 most active users from the whole trial were removed from the analysis.

The first section of the table gives various lift measures comparing the behaviour of each recommender group to the baseline group. These metrics reflect how much users act on recommendations, and are related to recall in the analysis of historical data. SIM-CF produced the best results on these metrics. The increase in open communication is important, since this is directly related to revenue. Even a small increase in this measure was considered significant from the business perspective. The second set of primary metrics (except lift in success rate) are also related to recall; these capture what proportion of the behaviour of users in the same group was produced by recommendations. Again SIM-CF performed the best, while ProCF+ showed the best lift in success rate, 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 the first set of metrics, 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 behaviour is much less, suggesting some "cannibalization" of search behaviour by the recommender, whereas in the other groups, the recommenders result in more additional user behaviour.

Whereas the primary metrics relate to short term user behaviour and revenue, the secondary metrics relate to more long term user experience, user satisfaction with the recommendations, and maintenance of overall user engagement. The interpretation of results is more subjective, since the metrics cannot be used to measure such properties directly.

The first such metric is the likelihood of a recommendation to lead to some reply (either positive or negative); the next metrics relate to contacts to highly popular users. The importance of these metrics is that many contacts, typically those to highly 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 this metric, all of our CF methods perform very well, since all are designed not to over-recommend highly popular users. 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 messages with a reply, the other CF methods are ahead on contacts to non-highly popular users. This may suggest that when highly popular users are recommended by SIM-CF, they are slightly more likely to generate a reply.

The second such metrics concern usage of the recommenders 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. Engagement of women is important to maintain a pool of contacts for the generally more active men. SIM-CF is the clearly best method for encouraging actions initiated by women.

During onsite testing of recommenders before the trial, the dating site customer service centre received several complaints from users about recommendations that were outside their preferred age range. Complaints were recorded against all recommenders except SIM-CF, with most complaints concerning Rules and RBSR-CF. The problem is that, while some users did not like recommendations not meeting their stated preferences, others did not mind and had successful interactions with them. The number of complaints was very small, but the issue was sensitive as it could affect perceived site reliability and user trust. Hence a filter based on dating site data analysis was implemented to all methods, limiting the age difference between the user and the candidate and allowing this difference to increase progressively with the user's age. This would add to the differences in performance between historical data analysis and trial results, especially for ProCF+ which was affected most. After the trial, some simple measures of age and location differences were calculated for 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 lift, 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.

5 Recommender Deployment

5.1 Initial SIM-CF Implementation

The implementation shown in Figure 1 was used for evaluation on historical data and in the trial. SIM-CF provides 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 used in the second stage of the process 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.

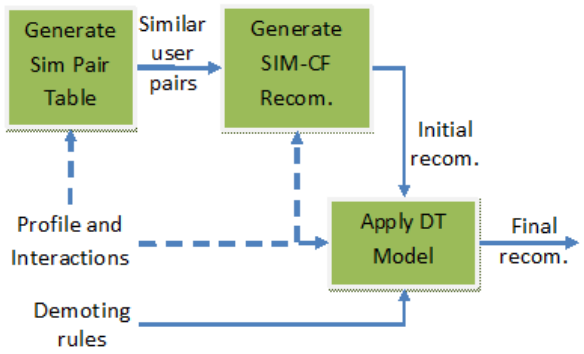

Figure 1: Generating SIM-CF recommendations

SIM-CF used an Oracle 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 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.

5.2 Deployment Process

The initial implementation of SIM-CF was high quality robust software, suitable for research and development and a rigorous trial for 10% of the users, but 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. This required a reimplementation of SIM-CF and integration with the production system. 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 in km between users and candidates. This allowed SIM-CF to provide more candidates, since, if the number of candidates in the same location as the user was insufficient, candidates further away could be used. Another change was to augment positive contacts 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.

Aug 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.

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

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

Nov 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 company indicated that the recommender system is stable and does not require any immediate additional maintenance directly related to the method itself. The Decision Tree rules have been tested against several datasets from different periods and showed 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 in the system.

6 Post-Deployment Evaluation

In this section, we compare the results from the trial and post-trial 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 Section 4.3 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 3 compares post-trial deployment metrics to those shown previously in Table 2 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 lift in the post-trial 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 (Section 5), which were made with a view to increasing contacts at the cost of slightly lowering success rate improvement.

The final section of Table 3 compares the trial and post-trial deployment values for the secondary 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.

Table 3: Comparison of trial and deployment metrics

	Trial	Deployment
Primary metrics: Comparison with baseline		
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%
Primary metrics: Usage of recommendations		
Lift in success rate over search	94.6%	88.8%
Proportion of contacts	11.8%	18.6%
Proportion of positive contacts	20.7%	30.2%
Proportion of communications	18.2%	28.3%
Secondary metrics: Usage of recommendations		
Contacts with no reply	26.1%	24.9%
Contact to non-highly popular	57.2%	59.3%
Positive contact to non-highly popular	62.0%	63.5%
Communication to non-highly popular	61.8%	63.8%
Contacts initiated by women	26.6%	30.6%
Positive contacts initiated by women	33.4%	39.3%
Communications initiated by women	28.2%	29.4%

We could not compare age and location differences for recommendations, as recommendations were not stored in the deployment setting. But since there were no complaints after deployment relating to age and location differences, we assume this aspect of the recommender is satisfactory.

7 Lessons Learned

Looking over the whole period of this project from inception to deployment, we identify several major lessons learnt 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: (i) 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 behaviour, (ii) commercial considerations go far beyond simple metrics used in the research literature, such as precision, recall, mean absolute error or root mean squared error, (iii) 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). We now elaborate on each of these points.

First, 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 behaviour *without* the benefit of the recommender (in our case, behaviour 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 behaviour using the recommender. Critical was the first trial (Krzywicki et al. 2012) where we learnt 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 paper 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 datasets. 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 an extended period of time with a changing user base.

Next, 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 tradeoffs 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 secondary metrics described above). The company chose the method for deployment by considering a range of metrics covering both short term revenue and user experience.

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.

In addition, we mention one limitation to our research. Since we were interested in improving overall user experience, we suggested a range of user interface questions to be explored, such as placement of recommendations, user control over recommendations, incorporation of live feedback from users, etc. However, any changes to the user interface of the production system would require lengthy design and testing, so it was impractical to investigate these issues.

8 Related Work

There are many deployed recommender systems, some of which are well known, such as Amazon and Netflix, however we are not aware of any other published work on a deployed, large scale people-to-people recommender system, although such recommenders are in use (e.g. Facebook and LinkedIn). There is some work published on recently deployed recommenders in a social networks context. Gertner, Richer and Bartee (2010) implemented a small scale (less than a few thousand users) recommender at MITRE Corp., called "Handshake", to help people find other users with similar interests and activities. Similar users are found by calculating the cosine or Jaccard score on user interests from various sources available on the web. The system is implemented as a web service designed to integrate with social networking sites.

A similar system, called "Do You Know?" (or DYK) for finding social contacts has been deployed by IBM Research

(Guy, Ronen, and Wilcox 2009) as a Widget on the IBM employee directory, called Fringe, showing some resemblance to the "People You May Know" widgets of Facebook and LinkedIn. A lot of attention was given to providing an explanation for each suggested contact, which was impossible in our setting. Over a period of four months it was used by over 6000 people, which is 40% of the site users. The acceptance rate of DYK was 60%, equal to that for Fringe.

Smith, Coyle and Briggs (2012) describe their experience of deploying HeyStaks, a social collaboration and recommendation system for sharing web searches. The queries of around 50,000 users were covered by Staks recommendations in about 50% of cases. They had to address "cold start" users (as in our case, these are new users joining the site) by promoting predefined results, before more specific recommendations could be generated for these users. In contrast, our method finds similar users by their profile and recommends their contacts to new users.

9 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 over 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% of active users clicked on recommendations and 33% of them communicated with recommended candidates.

If we are to list the main AI techniques that contributed to the success of the research, first, collaborative filtering allows recommendations to be based on user behaviour 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. Our research showed that no single AI method, whether Decision Tree learning, profile-based matching, or collaborative filtering, could alone produce satisfactory results. The best results were obtained by combining various techniques into one hybrid system.

Methods developed for this recommender 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 to reduce the recommendation frequency of highly 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

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 Pei, J.; Tseng, V.; Cao, L.; Motoda, H.; and Xu, G., eds., *Advances in Knowledge Discovery and Data Mining*. Berlin: Springer-Verlag.

Gertner, A. S.; Richer, J.; and Bartee, T. Q. 2010. Recommendations from Aggregated On-Line Activity. In *Proceedings of the 8th Workshop on Intelligent Techniques for Web Personalization and Recommender Systems*, 44–52.

Guy, I.; Ronen, I.; and Wilcox, E. 2009. Do You Know?: Recommending People to Invite into Your Social Network. In *Proceedings of the 2009 International Conference on Intelligent User Interfaces*, 77–86.

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 Kang, B.-H., and Richards, D., eds., *Knowledge Management and Acquisition for Smart Systems and Services*. 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 Anthony, P.; Ishizuka, M.; and Lukose, D., eds., *PRICAI 2012: Trends in Artificial Intelligence*. 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 Thielscher, M., and Zhang, D., eds., *AI 2012: Advances in Artificial Intelligence*. Berlin: Springer-Verlag.

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 Chen, L.; Triantafillou, P.; and Suel, T., eds., *Web Information Systems Engineering – WISE 2010*. 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.

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 Intelligentce (AAAI-02)*, 187–192.

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* 23:447–488.

Smyth, B.; Coyle, M.; and Briggs, P. 2012. HeyStaks: A Real-World Deployment of Social Search. In *Proceedings of the Sixth ACM Conference on Recommender Systems*, 289–292.