User Modeling and User-Adapted Interaction manuscript No.
(will be inserted by the editor)

Recommending people to people

The nature of reciprocal recommenders with a case study in online dating

**Luiz Pizzato · Tomek Rej · Joshua Akehurst ·
Irena Koprinska · Kalina Yacef · Judy Kay**

Received: date / Accepted: date

Abstract People-to-people recommenders constitute an important class of recommender systems. Examples include online dating, where people have the common goal of *finding a partner*, and employment websites where one group of users needs to *find a job (employer)* and another group needs to *find an employee*. People-to-people recommenders differ from the traditional items-to-people recommenders as they must satisfy both parties; we call this type of recommender *reciprocal*. This article is the first to present a comprehensive view of this important recommender class.

We first identify the characteristics of reciprocal recommenders and compare them with traditional recommenders, which are widely used in e-commerce websites. We then present a series of studies and evaluations of a content-based reciprocal recommender in the domain of online dating. It uses a large dataset from a major online dating website. We use this case study to illustrate the distinctive requirements of reciprocal recommenders and highlight important challenges, such as the need to avoid bad recommendations since they may make users to feel rejected. Our experiments indicate that, by considering reciprocity, the rate of successful connections can be significantly improved. They also show that, despite the existence of rich explicit profiles, the use of implicit profiles provides more effective recommendations. We conclude with a discussion, linking our work in online dating to the many other domains that require reciprocal recommenders. Our key contributions are the recognition of the reciprocal recommender as an important class of recommender, the identification of its distinctive characteristics and the exploration of how these impact the recommendation process in an extensive case study in the domain of online dating.

Keywords Recommender systems, Online Dating, Reciprocity

L. Pizzato, T. Rej, J. Akehurst, I. Koprinska, K. Yacef, J. Kay
Computer Human Adapted Interaction (CHAI), School of Information Technologies Building J12, The University of Sydney, NSW 2006, Australia. Tel.: +61-2-9351-3423 Fax: +61-2-9351-3838
E-mail: {firstname.surname}@sydney.edu.au

1 Introduction

Connecting people with other people is a central task in online social websites. Examples include finding friends, professional contacts, communities and people to follow on social networks; searching for partners and matching people in online dating websites; searching for jobs on employment websites and matching job applicants with employers; searching for mentors and matching mentors with mentees. With the exception of a few tasks such as recommending people to follow on Twitter[1] (Garcia and Amatriain, 2010), which focus on information sharing, people-to-people recommendations normally involve creating relationships that are *reciprocal*. In reciprocal relationships, both parties can express their likes and dislikes and a good match requires satisfying the preferences of both people. For instance, in the process of hiring someone for a job, both the candidate and the company offering the job need to assess each other; deciding whether the candidate is fit for the position and vice-versa.

In online dating, reciprocity is fundamental. Users will build a successful relationship only when both parties are interested in each other. Online dating websites provide the means for people to "meet" others virtually. This is in a safe environment, where they can exchange messages, before both users make the decision to meet each other face-to-face. Online dating is a domain that is attracting more and more users who otherwise would find it hard to meet people; in some cases, this is simply due to people's busy lifestyle (Reuters, 2010).

Reciprocity is a core requirement for systems that are designed to facilitate the establishment of reciprocal connection between people. Reciprocity, however, is a factor that standard information retrieval systems or recommender systems have not accounted for. With a few recent exceptions (Diaz et al., 2010; Pizzato et al., 2010b; Kim et al., 2010; Cai et al., 2011; Malinowski et al., 2006), most recommendation and information retrieval tasks consider one-sided relevance (i.e., whether the user receiving the recommendation has interest in the item being recommended). Although products such as movies and books may have particular target audiences that are more likely to benefit from them (e.g. the movie or book content would be better understood by the target group), sellers do not normally decide who can and cannot buy their products, and the actual products are not entities that care about who purchase them. Therefore, traditional one-sided search or recommendation strategies are sufficient for most of the current e-commerce needs.

In previous work, we first introduced the concept of reciprocal recommenders (Pizzato et al., 2010c). We identified some of their properties, used these to inform the design of a recommender and then evaluated the improvements obtained by taking account of reciprocity (Pizzato et al., 2010b). In this article, we draw upon the understanding we have built from these, and subsequent, studies to characterise the nature of reciprocal recommendations. We show how we have identified key hypotheses to explore and evaluated these in the context of a large data set for online dating.

1.1 Online dating

We now introduce the key concepts in reciprocal recommenders in terms of a scenario of a successful series of events in online dating:

1. User Bob signs up to the website and creates his profile;
2. Bob searches for potential dates, and finds Alice;

[1] `http://twitter.com`

3. Bob views Alice's profile;
4. Bob sends an expression of interest (EOI) message to her;
5. Bob receives a positive response from Alice;
6. Bob buys a token, enabling him to communicate with Alice and, from this point, he is allowed to send unconstrained messages to her;
7. Bob and Alice exchange contact details and their relationship is taken offline;
8. Bob and Alice are married and they live happily ever after.

Let us describe these events in more details.

During sign up (Event 1), users are requested to create a profile that specifies a variety of personal characteristics, using pre-defined data fields such as gender, age, location and education. Users are also asked to provide further details about their likes and dislikes, using their own words. Both pre-defined and textual information may contain characteristics of the sort of person they want to have a relationship with. This is an important part of the online dating process, because whatever a user discloses in their profile will affect their chances of a relationship.

A user profile for online dating is analogous to a curriculum vitae for employment websites. In both cases, a user profile is an incomplete, and possibly somewhat inaccurate or vague, representation of a user. However, as in a job interview, missing information is likely to be revealed when users begin to interact with each other. This gives people reason to provide rich and reasonably accurate profiles.

In order for a relationship to develop, users have to show mutual interest in each other (Events 4 and 5). Because unconstrained communications normally incur a monetary fee, online dating websites allow users to exchange EOI messages, which are short and predefined messages such as "I like you, do you want to talk?" The user receiving an EOI may decide to send a reply. This EOI reply can be either a positive reply (Event 5), indicating that the EOI receiver is also interested in the EOI sender, or a negative reply, indicating that the EOI receiver does not wish to receive further communication from the EOI sender. When an EOI receives a positive reply, we say that the interest is *reciprocated* and that the EOI was successful. Similarly, when an EOI receives a negative reply, we say that the interest is not reciprocated and that the EOI was unsuccessful.

After a positive exchange of EOIs between two users, they are offered the option to purchase tokens that allow them to communicate with each other using unconstrained messages (Event 6). These exchanges of unconstrained messages may evolve to an offline (outside the dating website) relationship (Event 7).

In this article, we limit our studies to the exchange of EOI messages and we focus on maximising positive responses to EOI messages (i.e. the number of successful interactions). The reason for this restriction is that the development of a relationship is uncertain after users have started to exchange unconstrained messages. For instance, two users might have liked each other at first, but did not emotionally connect when they met. Or, in contrast, other users might have taken a long while to decide to meet, but when they finally did meet, they really liked each other. People typically do not provide this type of information to the dating website; so it is not possible to measure levels of success at Event 8.

This article is organised as follows. Section 2 presents a literature review of the growing area of people-to-people recommenders. In Section 3, we present our analysis of the distinctive features of reciprocal recommenders. The next sections build upon these. Section 4 explores the criteria and measures for success in reciprocal recommenders. Section 5 reports on studies related to the effect of reciprocity, including the awareness of the need for reciprocity, the nature of the user profile, the effect of activity and popularity, and the effect of bad recommendations. Section 6 presents two other important aspects of the recommendation process – serendipity and transparency. Section 7 discusses the use of reciprocal recommenders for domains other than online dating and the issues for these. Finally, Section 8 presents concluding remarks.

2 Literature review

Connecting people with people is a central task in an important class of applications. The key role of reciprocity for recommending people to people has only recently being recognised. In just the last few years, there has been a growing interest in this area, with a number of papers presented at influential conferences. In this section, we review the previous work on connecting people with people in the following key reciprocal domains: social networks, mentor-mentee matching, job recommendation, and online dating. We also discuss algorithms for solving the stable matching problem.

2.1 Social networks

In the broad area of social matching, recommending people to other people (Terveen and McDonald, 2005) has a clear link with reciprocal recommenders because the quality of a match is determined by both parties involved in the match. However, some existing work on social matching tailors recommendations only to the needs of one party (Richards et al., 2008). Just a few papers mention the need for reciprocity and even fewer attempt to act on it.

IBM's enterprise social networking service, Beehive (Chen et al., 2009a), allows users to connect to friends and co-workers, post new information or comment on shared information. Two types of people recommender algorithms were compared: content-based and collaborative filtering. The content-based approach assumes that if two people post content on similar topics, they are likely to be pleased to get to know each other. It is based on similarity of textual content and uses content posted by the user on Beehive and additional information such as job description and location. The collaborative filtering is a typical friend-of-friend approach and uses only linking information from the social network. It is based on the intuition that if many of A's connections are connected to B, then A may like to connect to B too. The results show that all approaches increased the number of connections, compared to a control group that received no recommendations. The content-based approach was more successful in recommending contacts that were unknown to each other, while the collaborative filtering approach was more successful in finding known contacts. It is important to note that the befriending in Beehive is non-reciprocal, i.e. any user can connect with any other user without the consent of the other person. However, there are still important reciprocal social considerations as noted by the authors, e.g. before adding a contact, one has to consider how the other person would perceive this action and whether they will reciprocate the connection and also how the new contact will be perceived by the other people using the social network service.

Kim et al. (2010) created a people recommender system for a social networking website where users can reply positively or negatively to messages from other users. The authors distinguish between recommender systems for one-way interaction and two-way interaction. They propose an approach for a two-way interaction that considers both the interest of the sender and the interest of the recipient of message, and makes recommendations by combining them with a weighted harmonic mean. The method uses both user profiles and information about previous user interactions. For a given user, it finds the best matching values for every attribute and then combines them in a rule that can be used to generate recommendations. The success rate results show 36% improvement in comparison to the baseline success rate. The same research group also developed a collaborative filtering approach that was evaluated on same social networking website (Cai et al., 2011). The algorithm is called SocialCollab and considers the preferences of both sides. It is based on similarity of users in terms of attractiveness and taste. Two users are similar in *attractiveness* if they are liked by a common group of users, and these two users are similar in *taste* if they like a common group of users. To generate a recommendation for a user *A*, the SocialCollab algorithm considers all potential candidates *R*. For each candidate *R* it first finds two groups of similar users (in attractiveness and in taste); the candidate is added to the recommendation list for *A* if there is at least one similar user in both groups that reciprocally liked *A*. The recommendations are ranked according to the number of similar users. SocialCollab was shown to outperform standard collaborative filtering, confirming the importance of reciprocity in people-to-people recommenders. Cai et al. (2010) improved on these results by using gradient descent to learn the relative contribution of similar users in the ranking of the recommendations given by SocialCollab. In the same domain, the work of Kutty et al. (2012) have reported improvements over Cai et al. (2010) by using a model based on tensor decomposition to generate recommendations.

Fazel-Zarandi et al. (2011) studied different social drivers to predict collaborators in scientific collaboration networks. These social drivers include level of expertise, friend-of-friends, homophily, social exchange, and contagion. Fazel-Zarandi et al. found that these models could be used in combination to better predict collaborations, and that aspects such as homophily, and expert qualifications have a stronger impact in predicting collaborators than the network structure (including reciprocity). However, many of the social drivers may be considered reciprocal as aspects such as homophily, friend-of-friend and even level of expertise can be reciprocal (e.g. the mutually beneficial relationship between people students and mentors in scientific collaborations).

2.2 Mentor-mentee matching

The i-Help system (Bull et al., 2001) helped students find people who could assist them with university courses, e.g. first year computer science problems. A matchmaking system matched helpers with helpees by considering their attributes and preferences. For the helpers, it stored or inferred attributes such as knowledge of the topic, interests, cognitive style, eagerness to help, helpfulness, availability, and current load. The information was collected from several sources including self-evaluation and peer feedback in previous help sessions. An initial ranked list of potential helpers was produced. It was then refined by considering the preferences of the helpee, e.g. the importance of criteria such as helpfulness and urgency; the preferred and banned helpers. A final list of five potential helpers was compiled; the first of them to reply became the helper.

The PHelpS system (Greer et al., 1998) was an earlier prototype of i-Help. It was used in a workplace to train staff in how to use a new data management system. The candidate helpers were filtered based on their knowledge of the task, availability and load using a constraint solver. The list was presented to the helpee who chose the helper. Both i-Help and PHelpS relied on rich user models encoding the expertise and preferences of helpers and helpees.

2.3 Job recommendation

Malinowski et al. (2006) investigated the problem of matching people and jobs and argued that the matching should be reciprocal, considering the preferences of both the job seeker and the recruiter. They built two recommender systems. The first one recommended job seekers (i.e. their resume/profiles) to job descriptions of a particular recruiter. To create training data, a recruiter manually labelled the resume of a set of people as either fit or not fit for a list of jobs. The attribute set included demographic, educational, job experience, language, technology skills and other attributes. The second system recommended jobs to job seekers. To create training data, the job candidates were asked to rank a set of job descriptions indicating how well the jobs fitted their preferences. In both cases the Expectation Maximisation algorithm was used to build the prediction model. The two recommender systems were evaluated separately and showed promising prediction accuracy results. Several methods for combining the two recommendations were proposed but were not implemented and evaluated. More broadly, methods for combining different recommenders have been summarised by Burke (2002).

2.4 Online dating

Despite the popularity of online dating services, little research has been reported on building recommender systems for online dating. However, the interest is growing and several papers have been published recently, e.g. Pizzato et al. (2010b), Diaz et al. (2010), and McFee and Lanckriet (2010).

One of the first studies of recommender systems for online dating was conducted by Brožovský and Petříček (2007) who evaluated two collaborative filtering based approaches (item-to-item and user-to-user). A data sample from a commercial dating website was collected, where users rated the attractiveness of other users based on their photos. The predictive accuracy of the collaborative filtering algorithms was evaluated and the results showed that both algorithms outperformed the baselines. The authors mentioned the need for reciprocity, but did not explore it.

In Pizzato et al. (2010b), we proposed a content-based system which used both user profiles and user interactions. To produce recommendations for a given user, it extracted his/her implicit preferences (i.e. the preferences that are inferred from the interactions with the other users) and then matched them with the profiles of the other users. We showed that reciprocity improved both the success rate and recall of the recommender. In Pizzato et al. (2010a), we found that the implicit preferences produced better recommendation than the explicit preferences (i.e. the preferences stated by the user), provided that there is enough previous interaction data to infer the implicit preferences. More recently, we proposed a recommender system for online dating that combined content-based and collaborative filtering approaches and utilised both user profiles and user interactions (Akehurst et al., 2011).

Diaz et al. (2010) formulated the matchmaking task as an information retrieval problem, where user profiles were ranked with respect to a given ideal partner profile (i.e. explicit user preferences). Using historical data, a training set of matches (pairs of users represented with their profile attributes) was created and labelled as relevant and non-relevant. A match was considered relevant if users exchanged contact information, and irrelevant if one of the users inspected the other's profile but did not send a message or if he/she sent a message but the other user did not reply. A machine learning classifier (ensemble of boosted regression trees) was built and used to predict the relevance of new matches; given a new user, the potential candidates were ranked based on their predicted score. The approach was evaluated using data from an online dating website. The authors described the reciprocal aspect of their work as two-sided relevance and stressed its usefulness for ranking candidates in matchmaking problems.

McFee and Lanckriet (2010) proposed an approach that learnt distance metrics that were optimised for different ranking evaluation measures, e.g. mean average precision and area under the curve. The metric learning task was cast as an information retrieval problem using a machine learning algorithm (structural support vector machine) to learn the metric, given a ranking. The method was evaluated using data from an online dating website. Similarly to Diaz et al. (2010), each training example was a pair of users represented with their profile features and labelled as a successful or unsuccessful match (the match was successful if the users had expressed mutual interest and unsuccessful otherwise). The results showed that the new method was slightly better than the baseline (Euclidean distance measure). Reciprocity was not discussed in the paper; its main focus was the new general algorithm for learning distance metrics rather than the online dating application.

2.5 Stable matching

The stable marriage problem deals with the problem of finding matches between men and women such that there are no two people, who are of opposite gender and who would rather be matched with each other than with their current partners (Gelain et al., 2010). This problem was formulated by Gale and Shapley (1962), who proposed an algorithm to solve it in polynomial time. Gale and Shapley worked in the domain of college admissions, where both colleges and students have preferences for each other. Since then, several similar problems under the name of stable matching problems have been defined. For instance, in a realistic dating scenario people do not express their preferences fully, and these might have ties. Due to different types of sexuality, the dating network is not bipartite and matching pairs might be polygamous. Two similar stable matching problems that have some of these real world characteristics are the roommate problem and the intern assignment problem (Ronn, 1990). Stochastic matching is a similar matching problem that was motivated by online dating and kidney exchange programs where graph edges have probabilities of occurrence (preferences) and each node contains a patience value indicating how long the node is willing to endure bad matches (Chen et al., 2009b; Bansal et al., 2010).

In recent work (Pizzato and Silvestrini, 2011b), we built a framework for testing stochastic matching algorithms in the recommendations given by a collaborative filtering approach. Our initial experiments, with a greedy stochastic matching algorithm showed that stochastic matching can ensure an even number of recommendations for all users and improve their top-N recommendations.

2.6 Summary

In summary, we can draw upon an existing understanding of some of the distinctive aspects of reciprocal recommenders in various domains. At one level, the concept of reciprocity in recommender systems is not new; it has appeared under different names such as “two-sided”, “bilateral” and “two-way” interactions. At the same time, we have not found other work that has recognised, defined and explicitly explored the distinctive characteristics of reciprocal recommenders, how these impact the effectiveness of various recommendation approaches

Table 1 Terminology for reciprocal recommenders

Definitions for people and their roles

- *subject* : user seeking recommendations
- *object* : person being sought by the subject (analogous to items for traditional recommenders)
- *proactive user*: user who tends to take the *subject* role
- *reactive user*: user who tends to take the *object* role

Definitions associated with acquisition of the user model

(i) *explicit user profile*: properties provided by the users about themselves – may be visible on the website
(ii) *explicit preference model*: properties stated by the users about the object they are looking for – may be visible on the website
(iii) *implicit preference model*: properties inferred, from the user's interactions, about the objects they like and dislike – can remain internal to the system or be revealed to the user
- *user model*: internal model built and maintained by the system about the user's characteristics, preferences and history of interactions – uses (i), (ii), and (iii).

Definitions associated with user actions

- *EOI*: expression of interest, sent by the subject to an object indicating that the subject likes the object;
- *EOI response*: object response to the subject's EOI (either *positive* or *negative*).
- *connection*: action that occurs after a recommendation has been accepted by *both* subject and object. In online dating the subject accepts the recommendation by sending an EOI to an object and the object sends a positive EOI response.

and how this informs the approaches that are likely to be most effective when creating such recommenders. This paper builds upon our previous, partial definition of reciprocal recommenders, to provide a comprehensive and systematic base for designing them. We further use the definition to drive the principled evaluation of approaches, which we report in the context of a large dataset from a commercial online dating website.

3 The distinctive nature of reciprocal recommenders

The reciprocal recommender performs the important task of helping people find other people in a particular context. Although the exact definition of reciprocity is context dependant, reciprocal recommenders focus on satisfying the preferences of both parties in the recommendation. Given the importance of the task, we analysed the distinctive aspects of reciprocal recommenders, compared with traditional product-based recommenders, which have dominated recommender research so far. This is a foundation for identifying the key research that cannot be taken directly from the established recommender research and for defining the areas that still need to be explored.

3.1 Terminology

We need to introduce terminology that helps distinguish and characterise reciprocal recommenders. This is summarised in Table 1.

A traditional recommender aims to find a set of *items* to recommend to each user. In reciprocal recommenders, where the goal is to recommend people to people, items are also users. We distinguish the roles of the users in the recommendation process. We use the term *subject* for the user who seeks recommendations; the term *object* refers to the person who is recommended to the subject. This means that for a reciprocal recommender, one person can be a subject in one set of recommendations and an object in the set of recommendations to other users.

Although an individual may take the role of subject or object, in practice individuals may tend to take primarily one of these. For example, for online dating, we found that some users are *proactive*, often playing the subject role and initiating requests for recommendations; this is a stereotypic male behaviour. Other people tended to be *reactive*, registering with the site and waiting to be approached, primarily taking the object role. (Stereotypically, this is a female role.) Our definitions allow for the possibility that a person may change roles at different times. In some reciprocal recommenders, the roles may be more fixed. For example, in employment websites some employers may post jobs and wait for potential employees to respond; in this case, the employers are reactive and employees proactive. In online dating, some individuals maintain a fixed proactive or reactive role, while others do not. Although the particular domain affects the potential role an individual will take, the existence of these different roles has the potential to affect the design of effective recommender algorithms.

In Table 1, we also present terms related to the creation of the user model within reciprocal recommenders. Explicit user profiles and preferences are provided by the people using the system, while the implicit preferences are inferred based on people's actions. The user model is created from both the explicit and implicit sources of information. For online dating, the *explicit user profile* is the information the users provide about themselves. This is typically a combination of answers to closed questions as well as free text information with self descriptions and a photograph. The *explicit preference model* is the information the users provide about the person they are seeking. The final source of user modelling evidence comes from the *implicit preference model* which the recommender system builds, based on the user's actual activity on the site. For an employment web site where employers post advertisements and await applications from potential employees, there are two cases. For the job seeker, the explicit user profile is their curriculum vitae, the explicit preference model is the details of the jobs they are seeking. For the employer, the explicit user profile is the information they provide about the job and employer, and the explicit preference model is the details of desired employees.

The user's implicit preference model is built using the user's actions within the website. These actions indicate the objects the user likes and objects they dislike. The online dating website behind the success story between Bob and Alice described in Section 1 use several pieces of evidence to define Bob's and Alice's preferences. For instance, taking Bob's proactive point of view: he viewed Alice's profile, he decided to send her an *EOI*, and he purchased a token to be able to talk to her. Similarly from Alice's reactive point of view: after she received the EOI from Bob, she may have viewed his profile, and sent Bob an *EOI positive response*. Once this was done, Bob and Alice established a *connection*, which is a measure of the effectiveness of a recommendation method.

In practice, in addition to EOIs, several other actions indicate preferences between subjects and objects. For instance, if a subject spends some time viewing the profile of an object, this may be interpreted as a weak positive indication of preference; however, if after such a careful view of the profile, the subject does not perform any other action (i.e. the subject does not send an EOI to the object), then the view may instead be interpreted as a weak negative indication of preference.

A similar, but somewhat different case can occur if, after receiving and reading an EOI, the object decides not to respond to it. This can be seen as an implicit indication that the

Table 2 Comparing traditional and reciprocal recommenders for the creation of the user model

	Traditional Recommenders	**Reciprocal Recommenders**
[a]	Users have no reason to provide explicit user profiles.	Users expect to provide detailed self-profiles.
[b]		Explicit *user profiles* may contain inaccurate elements.
[c]	Users typically prefer to provide minimal details of preferred objects.	Users expect to provide detailed profiles of their preferred objects.
[d]		Explicit *preferences* may contain inaccurate elements.
[e]	Satisfied users are likely to return for more recommendations.	Many users leave the system after a successful recommendation.
[f]	Because of their long history using the recommender, users have rich implicit preferences.	Less common for users to have rich implicit preferences.

object does not like the subject. Taking an employment website example, suppose a person seeking a job sees one they like and submits their CV. Suppose the potential employer reads this and decides not to respond at all. The candidate will not receive a negative response to their application, but the candidate cannot continue along the selection process to an interview.

3.2 Comparison between traditional and reciprocal recommenders

We have identified the distinctive aspects of reciprocal recommenders in terms of the creation of the user model, the role of the people using the system and success measures. We summarise these in Tables 2, 3, 4, and 5.

Table 2 focuses on user profiles and preferences. In traditional recommenders, there is typically no need for explicit user profiles and users are unwilling to provide detailed explicit information about what items they prefer. Sometimes, users may not be aware of their specific needs and wants. For instance, Agrawal et al. (2009) stated that users may not know what they want when reading news websites. Blecker et al. (2004) report that consumers are not aware of their preferences when purchasing products until they are violated. Therefore, asking users to define their explicit user profile and preferences in a traditional recommender domain may not give reliable information. It is also undesirable because it forces a burden on the user.

By contrast, in reciprocal recommenders, people normally expect to provide a detailed explicit user profile of themselves and of their preferences. Importantly, this explicit information may be inaccurate. In the case of the explicit user profile, this may be due to a lack of self-awareness, or to the desire to have a more attractive profile – even if it is not entirely accurate. For explicit preferences, users may also lack self-awareness or have difficulty formulating their real preferences for objects. These differences have important implications that should influence the design of a recommender. Notably, the implicit preferences, once available, may be used to find inaccuracies in the explicit preferences. It may also be valuable to present these inaccuracies to the users, so that they can decide whether they wish to change their explicit preferences. A recommender needs to determine how to make best use of conflicting evidence from explicit and implicit preferences. Research is still needed on this aspect.

Table 3 Comparing traditional and reciprocal recommenders on the user role

	Traditional Recommenders	**Reciprocal Recommenders**
[g]	Success is determined solely by the user seeking the recommendation.	Success is determined by both subject and object.
[h]	User receiving recommendation is aware that they are the sole arbiter of its success.	User (the subject) is aware that success also depends on the other person involved (the object).
[i]	Users are usually the proactive seekers of recommendations.	Users can be either proactive when in a subject role or reactive when taking an object role.

The last two rows of Table 2 deal with the creation of implicit profiles based on the activity of users at the recommender site. Successful traditional recommenders are likely to have loyal users who repeatedly use the site, so allowing it to build rich implicit preferences. By contrast, the nature of reciprocal recommenders means that people may leave the site permanently after a successful recommendation. For example, a person who uses a dating site to find a lifelong spouse and succeeds may have no need for the site thereafter. A similar case applies when a candidate accepts a long term job position.

Table 3 deals with the nature of the user's role. In traditional recommenders, the user is the sole arbiter of the success of a recommendation. For reciprocal recommenders, success depends on users on both sides of the recommendation. As reciprocal recommender users are aware of this, it can affect their behaviour. For instance, users may not accept a perfectly good recommendation (someone who is very attractive to them) because they may fear they will not be liked by the other person.

The last row of Table 3 notes that recommender research has been dominated by contexts where the users are proactive and the items are passive. In reciprocal recommenders, users can be both proactive or reactive. This means that when the subject receives a recommendations (the object), they will have to analyse it to decide whether they like the object and if they predict that the object will reciprocate, and like the subject. When the reactive user (the object) receives an EOI originating from a recommendation, they already know that the subject likes them. Therefore, the object's decision to show interest in a subject is simply based on their own preferences.

Table 4 explores differences in the nature of success for reciprocal recommenders. Row [j] reflects the fact that success in recommender research has been heavily based on the *accuracy* of the recommender, where this is assessed in terms of the user's response to recommendations. The steps in the reciprocal recommendation process provide three important and natural places to measure accuracy. First, taking the perspective of the subject, one can analyse how many times subjects accepted the recommendations given to them. Second, from the object's point-of-view, the analysis should assess how often they have been accepted in a recommendation. Finally, from the connection perspective, we can measure how many connections were formed after the recommendations were given.

Rows [k–n] relate to spread and coverage of recommendations. While some recommender research has considered these issues, they are central to judging the success of a reciprocal recommender. In particular, a set of poor recommendations may result in a person receiving several negative responses to their EOIs; this can be very discouraging and upsetting. Row [m] points to the need to avoid recommending the same person too often as that person may be overloaded. This is undesirable as such people are more likely to reject EOIs, increasing the risk of rejection for those recommended to the popular and overloaded person. By contrast, it may be perfectly acceptable if a traditional recommender presents the same item, such as a book, to a very large number of users. Row [n] points to the need for a

Table 4 Comparing traditional and reciprocal recommenders for the evaluation and success criteria

	Traditional Recommenders	Reciprocal Recommenders
[j]	Success is often based on accuracy, in terms of user response to recommendations.	Success measures can be taken from three different perspectives: the subject, the object, and the connection between subjects and objects.
[k]	A small proportion of poor recommendations is acceptable.	Poor recommendations are to be avoided because users can suffer if they are rejected repeatedly.
[l]	Users expect a large number of good recommendations; therefore, they are likely to accept a few bad recommendations.	Users are looking for a small number of very good recommendations (often just one); a recommendation system that gives even a few bad recommendations may be very discouraging.
[m]	Typically, it is acceptable to recommend the same items to many users.	The same object must not be recommended to too many others.
[n]	Typically, it is acceptable *not* to recommend some items at all.	Proactive users may not need to be the object of recommendations; however, every strictly reactive user should be recommended to other users.
[o]	Serendipity may be accepted and even desirable in some recommenders, where users do not provide their explicit preferences.	Users may be disappointed when recommended objects that do not match their explicit preferences, implying the need for good interface design.

successful reciprocal recommender to ensure that every reactive person is recommended to some users. This is in stark contrast with traditional recommenders which may not recommend some items at all, for example, if they are genuinely undesirable.

Row [o] notes the potential differences in the value of serendipity. Since the users of a reciprocal recommender have specified their explicit preferences, the recommender interface might need to present justifications for failing to match the person's stated preferences.

3.3 Research directions for reciprocal recommender algorithms

Based on the fundamental differences we have identified between reciprocal recommenders and traditional recommenders, we now consider the implications for the recommender algorithms. Table 5 shows the three main approaches to traditional recommender systems for product recommendations and some differences when they are applied to reciprocal recommendation.

Reciprocal recommenders are distinctive in starting up with a rich profile for each subject and their explicit model for objects. This means that *content-based* approaches can operate immediately to generate object recommendations. This contrasts with traditional recommender systems where users do not want, or need, to provide rich user models for good recommendations. The lack of rich user content is one of the reasons why traditional recommenders make use of techniques such as collaborative filtering, which does not need content descriptions for either users or items.

Collaborative filtering recommends items that the user has not seen, based on the intuition that this user will like the same objects that are liked by users with similar behaviour. When collaborative filtering is applied to a reciprocal domain, the definition of similar behaviour is less clear. Reciprocal recommenders should make use of EOIs, responses and connections between subjects and objects. For instance, if a subject likes a recommended object and sends an EOI, success depends on the response of that object. A collaborative filtering method can ignore the response and implement a collaborative filtering method that recommends to subject A all objects B whose recommendations contain A (Pizzato

Table 5 Comparing traditional and reciprocal recommenders for the recommender algorithms

	Traditional Recommenders	**Reciprocal Recommenders**
[p]	Because users seldom provide self profiles, *content-based* approaches are normally limited to the item's content.	Because explicit user profiles, including explicit preferences, are normally available, *content-based* approaches can make use of content for both subjects and objects.
[q]	*Collaborative filtering* looks into the purchasing history of users and by finding groups of users who purchased the same set of items, it recommends items to these users that have not yet seen them.	Reciprocity means that *collaborative filtering* cannot simply use classic one way preferences.
[r]	*Hybrid* algorithms, combining content-based and collaborative filtering, can overcome the new user and the new item problems.	Because users need good recommendations from the start and they might require just one successful recommendation, it is very important to give good recommendations for new users, making *hybrid* approaches important.

et al., 2010c). On the other hand, it might be important to consider that similar users may receive similar responses (both negative and positive). In this sense, some collaborative filtering techniques might only use positive responses to ensure reciprocity (Cai et al., 2011). Another possible strategy could take account for additional factors that are important. For instance, users who are predicted to be more likely to respond positively to messages (as in Akehurst et al., 2011) may make poor recommendations if those subjects have already found their match. For the same reason, it may be beneficial to recommend objects who normally reply negatively to EOIs.

Hybrid approaches that combine content and collaborative filtering techniques are used to overcome limitations of both techniques. When new users have not been involved in any interaction, the new user problem cannot be solved by looking at interactions alone (as in collaborative filtering techniques). Therefore, the use of content information from both the subjects and the objects can help minimise the cold-start problem.

4 Definition of success and evaluation metrics

Success is obtained when a certain goal is accomplished. Different recommender systems have different goals. For instance, a book recommender is successful when the customer buys a book; a personalised web advertisement engine is successful when a user visits the advertised web page; an online dating recommender is successful when users are connected and decide to date each other.

Row [g] of Table 3 differentiates reciprocal recommenders and traditional recommenders in the way success is defined. Since success for reciprocal recommenders involves both sides of the recommendation, one cannot assess success with just a single sided preference or relevance metric. This is highlighted by Diaz et al. (2010) as a two-sided relevance score. In Pizzato et al. (2010b), we have found a considerable improvement in the recommendation set when considering the preferences of both the subject and the object of the recommendation.

4.1 Reciprocal recommender evaluation

Because of the high cost involved in performing online evaluations of recommender systems with live users, it is important to conduct offline evaluation using historical data (Gunawardana and Shani, 2009). Historical data contains all successful and unsuccessful interactions of the users within the recommender domain. The success of a recommendation can be either binary (successful or unsuccessful) or can be represented as a scale of success (e.g. a movie recommendation that did not lead to a purchase, but because the user had already seen the movie and liked it, the user wrote a positive review).

For reciprocal recommenders, the evaluation can also use different levels of success. For instance, a job recommendation can be seen as somewhat successful when the user (as a subject) decides to apply for the recommended job; the same recommendation is more successful when the same user is called to be interviewed for the recommended job (the object); and even more successful when the user is selected for the position.

In order to define one metric that covers all the levels of success, one must quantify the importance of each of these levels in relation to the others. To determine such a measure for an employment website, it is necessary to assess the relative importance of a user being selected for an interview, versus a user applying for a job. Setting precise weights for the importance of the various success measures is difficult; it may depend on business strategies and these might change over time. A more tractable evaluation strategy is to define different metrics for different levels of success, in order to understand how the recommendation algorithm and its parameters influence the various phases of the recommendation process. In online dating, for instance, consider two definitions of success: (1) *like*, indicating whether the subject likes an object; and (2) *like-back*, which represents a reciprocated recommendation (a subject likes an object, who also likes the subject). In this article, we particularly focus on the *like-back* relationship between two users, because it directly correlates with the site revenue and the user satisfaction.

Non-reciprocal success is normally less useful for reciprocal domains since the goal of the recommendation is to maximise a reciprocal relationship. However, non-reciprocal success measures can explain some user behaviour, such as whether the user is considering the reciprocal aspects of the domain or not. Traditional recommenders might have different levels of success (e.g. view, buy, review a product), but none of them reflect reciprocal success.

Historical data can show the success or failure only for events that actually occurred. So there are many potential recommendations for which there is no information. This makes it important to consider measures that account for the number of *successful* recommendations as a proportion of all recommendations and as a proportion of all *known* successful and unsuccessful recommendations. These measures are *precision* and *success rate*. Since it is also important to evaluate recommenders in terms of their performance with fixed numbers of recommendations, we define *precision at n* ($P@n$) as the proportion of successful recommendations for a set of n recommendations; this definition is consistent with the definition of precision used in information retrieval.

$$P@n = \frac{\#\ \text{successful}}{n\ \text{recommendations}} : \text{given } n \text{ recommendations} \tag{1}$$

Similarly, the *success rate at n* ($S@n$) is the ratio of the number of successful recommendations and the *known* (successful and unsuccessful) recommendations in the first n

recommendations.

$$S@n = \frac{\#\ \text{successful}}{\#\ \text{successful} + \#\ \text{unsuccessful}} : \text{given } n \text{ recommendations} \quad (2)$$

Unsuccessful EOIs are those that did not receive a positive reply. However, in some cases, EOIs might not be read by their intended recipients. EOIs that have been read and received a negative replied are said to be failed. *Failure rate at n* ($F@n$) can tell us whether a ranking strategy helps minimising negative responses if the first n recommendation were accepted by the users. Therefore, $F@n$ is particularly important for evaluating user satisfaction which is a key aspect of online dating systems. This is consistent with Gunawardana and Shani (2009) who emphasised that in online dating minimising the false positive rate is more important than maximising the true positive rate (precision).

$$F@n = \frac{\#\ \text{failed}}{\#\ \text{successful} + \#\ \text{unsuccessful}} : \text{given } n \text{ recommendations} \quad (3)$$

When using historical data, it is also possible to assess *recall at n* ($R@n$), which indicates how close a recommendation set is to containing all known successful interactions. Recall is important in many information retrieval contexts; for recommender systems recall can tell about the reliability of the other performance metrics that rely on historical data, but it cannot tell how well the performance of a live system would be.

$$R@n = \frac{\#\ \text{successful}}{\#\ \text{all known successful interactions}} : \text{given } n \text{ recommendations} \quad (4)$$

A more useful metric for recommender systems is *coverage at n* ($C@n$). This is the proportion of all users who could be recommended (i.e. an object) given the recommendations for a set of subjects.

$$C@n = \frac{\#\ \text{distinct objects}}{\#\ \text{distinct subjects}} : \text{given } n \text{ recommendations} \quad (5)$$

Other information retrieval measures have been considered to evaluate relevance of a retrieved set of documents. Diaz et al. (2010) used average precision (AP), $P@n$, normalised discounted cumulative gain at rank k (NDCGk) and expected reciprocal rank. McFee and Lanckriet (2010) used area under the ROC curve, $P@n$, AP, mean reciprocal rank and NDCGk.

5 Studies of elements of reciprocity

This section reports our studies of the elements identified in Tables 2, 3, 4, and 5. We begin with work that explores how we designed a recommender that takes account of reciprocity and its evaluation. We then report studies that confirm our claims of the importance of people's awareness of the need for reciprocity, the nature of the profiles, analyses of user activity, the impact of user popularity, and weighting the preferences of one side over the other. We also study the need to reduce the risk of people being rejected due to unsuitable recommendations where subjects would accept recommendations of objects who will later reject them.

Our studies were in the context of a large online dating site. In the remainder of this section, we report several studies. Some evaluated algorithms to assess the effect of recipocal recommenders and reducing the risk of rejection. For these evaluation experiments, we used the users' contact history over a six week period; where the first four were used as training data and the remaining two were used for testing. In addition, the users' profile information was used to train the content-based recommender and to find matching users. The training set had 1.4 million messages sent by over 90,000 users. Note that we are only testing subjects and objects who have sent or received a message during the training period. While we report the results for a particular six week period, we also repeated the experiment over other periods, with near identical results.

5.1 The effect of reciprocity

As described in Row [g] of Table 3, success of reciprocal recommenders depends on agreement by both parties in a recommendation. In Pizzato et al. (2010b), we designed RECON as a content-based recommender that considers reciprocal preferences to recommend users to each other. RECON's reciprocal recommender process is described in Algorithm 1 and its compatibility scores in Algorithm 2. In Algorithm 1, the reciprocal score for a pair of users is calculated as the harmonic mean of the scores from one user to the other and vice-versa. We used a harmonic mean because we achieved best results when the reciprocal score was between the largest and the lowest individual compatibility scores, and closer to the lowest score. The harmonic mean guarantees not to assign a large score for users whose preferences towards each other differ considerably, in terms of compatibility scores.

Algorithm 1: ReciprocalRecommender(x,N)

```
Input: User x, N number of recommendations to provide
Output: List of (Recommendations R, their Scores S
begin
    Find P_x
    R ← M_x^c                                   // all users not messaged by x
    S ← {s_y : ∀y ∈ R}
    foreach y ∈ R do
        s_y ← Compatibility(P_x, y)
        if s_y > 0 then
            Find P_y
            /* Calculate the reciprocal score as the harmonic mean of the two reciprocal
               compatibility values                                              */
            s_y ← 2 / ((s_y)^-1 + (Compatibility(P_y, x))^-1)
    R ← {y_1, y_2, ... y_j : s_{y_i} ≥ s_{y_{i+1}} ∀i}      // sort by reciprocal score
    if ∃N then
        R ← R \ {y_k : k > N}
    return (R, W)
end
```

Our empirical evaluation showed both success rate and recall improve when the algorithm takes account of the preferences of both parties in the recommendation Pizzato et al. (2010b). The improvements gained by taking account of reciprocity are presented in Figure 1, which compares $S@n$ and $R@n$ for different values of n for a reciprocal recommender (RECON) and for a non-reciprocal recommender[2] on the same set of users (subjects and objects). This evaluation is also compared to a baseline success measure (17.3%) which

[2] RECON without the reciprocal score; where all recommended objects y to subjects x are based solely on how much the subjects x like the objects y (the likes of the objects y are ignored).

Algorithm 2: $Compatibility(P_x, y)$

Input: Preferences P_x, user y
Output: Compatibly score s
begin
  /* Return a small non-zero value if x has no preference (e.g. x did not send any message) */
  if $P_x = \emptyset$ **then**
    return 0.001
  else
    Extract user y's profile U_y
    foreach *Attribute* $a \in A$ *list of attributes* **do**
      Obtain the value v_a of attribute a in U_y
      Get $p_{x,a}$ in P_x Find $(v,n) : (v,n) \in p_{x,a}, v = v_a$
      if $n = 0$ **then**
        /* e.g. male user y for a strict heterosexual male user x) */
        return 0
      else
        /* s_a is the percentage of the distribution that is equal to v */
        $s_a \leftarrow \frac{n}{\sum n \in p_{x,a}}$
    return $\frac{\sum s_a}{|A|}$ // Arithmetic mean of all s_a
end

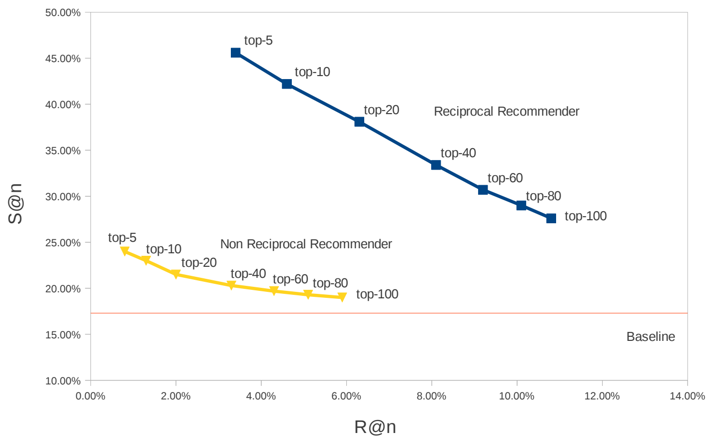

Fig. 1 Success rate and recall comparison between reciprocal recommender, non-reciprocal recommender and the current search made by users (baseline)

reflects the proportion of messages sent by users who get positive replies in the testing set. The baseline measure reflects how well users did, unaided by a recommender, in their search for people they like (and who might like them).

These results indicate that even using only the user implicit preferences, the results are better than the baseline. The baseline success rate, can be seen as the success rate of a set of random recommendations sampled from the user's set of "liked" profiles. It can also be noticed that RECON shows a large improvement in both $S@n$ and $R@n$. For the list of top-10 recommendations, the reciprocal recommender has a success rate of 42.20% while the non-reciprocal recommender has 23.00% success: an improvement of 83.48% due to the use of reciprocity. Similarly, for the top-100 recommendations, the addition of reciprocity

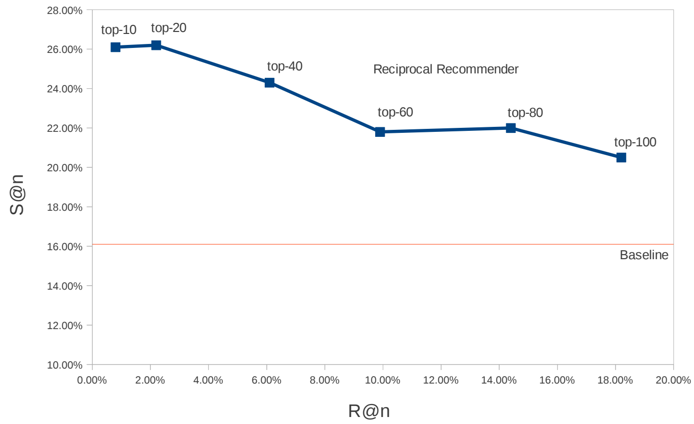

Fig. 2 Success comparison between reciprocal recommender and the current search (baseline) made by new users (cold-start problem)

improved recall by 83.05% (from 5.90% to 10.80%). This supports our hypothesis that reciprocity is important for people-to-people recommender systems where both parties must be satisfied (i.e. reciprocal recommenders).

Because a user model can be computed for each individual user, it is possible to provide personalised recommendations for each user. For those users whose preferences are not (yet) available, we can assume that they will equally like all users and find those who will like them back.

In Pizzato et al. (2010b), we explored the cold start problem, the case of new users. We observed that 16.10% of messages sent by new users received positive replies, indicating that they have a slightly lower success rate than the 17.3% for all users. In Figure 2, we show the success rate of RECON for these new users, compared with the baseline. The figure shows that RECON predicts with higher accuracy whether a communication will be successful. In Figure 2, we observe that for the top-10 and top-20, one in four known messages are successful. This performance is achieved by pre-filtering the list of possible recommendations using the preferences of the new user's peer group. This is an assumption that has to be made, otherwise too many users would have to be ranked and the runtime performance of the system would be harmed considerably.

5.2 Success awareness

Row [h] of Table 3 highlights how users are well aware of the need for reciprocity for success. In online dating, subjects know that they cannot start a communication based only on the qualities they like, and that they must chose an object who they think will like them back. Therefore, it is very likely that subjects paying for a communication are influenced by the extent to which they match the object's preferences. To test this, we compared the proportion of views, EOIs and paid communications where subjects matched their objects' preferences versus the other way around.

Table 6 Proportion of views, messages and paid communications done by a subject towards an object that the object matches the subject's explicit preferences and that the subject matches the object's explicit preferences

Action	views	EOIs	paid communication
Number of actions	39,016	5,692	864
Object matches subject's preferences	27,270 (70%)	4,453 (78%)	650 (75%)
Subject matches object's preferences	21,146 (54%)	3,228 (57%)	578 (67%)

Table 6 shows the proportion of the communication among users where they "match" each other's explicit preferences. A match consists of one user having all profile features that were selected in another user's explicit preference. For example: Alice matches Bob's explicit preferences because Bob's explicit preferences are defined by the features "female", "red hair", and "blue eyes", all of which are defined in Alice's explicit user profile.

We observe in Table 6 that the subjects tend to follow their preferences more closely (70%) when viewing the objects, and they do not have much consideration for the object's preferences (54%). Because profile views occur before the subject is able to assess whether the potential date matches his/her preferences, the percentage of matching between the object's preferences and the subject's profile is slightly above random and likely due to general group preferences (e.g. similar age, location, height). However, the proportion of reciprocal interest increases when the subject is sending EOIs and paid communications[3] (they consider the object's preferences, 57% and 67% respectively). In contrast, the proportion of people following their own preferences is fairly stable across the activities (75% of paid communications have an object matching the subject's preferences). Therefore, the increase shown in this matching for the two types of communication indicates that subjects consider the reciprocal aspects of the domain before engaging more seriously with objects.

5.3 User profile

Section 3 argued that a characteristic of reciprocal recommender domains is that users expect to provide detailed self-profiles (Row [a] of Table 2) and to provide detailed profiles of their preferred objects (Row [c]). Users should see a clear advantage in disclosing this information as it can be used for searches (when behaving proactively) and should help others find them (when behaving reactively).

We explored this by analysing how profile completeness (measured as the percentage of predefined fields in the user's profile that were filled) relates to the average number of EOIs a user receives. We selected a population of well-established, active users: those who had been active on the website over a period of 15 to 30 days, and who had sent or received a normal range of EOIs (e.g. sent or received at least one EOI and no more than 200) in the period while they were active. We observed that once users' profiles were over 65% complete, they averaged more EOIs than the others (statistically significant $p < 0.01$). At the 65% complete level, subjects averaged 30 EOIs, while those with profiles 40% complete averaged 15. This confirms the advantage of more complete profiles for users, as they will receive more attention from the other users. Similarly, users with longer textual profiles are both more engaged with the website and receive more communication from other users (also statistically significant $p < 0.01$).

[3] As stated on page 3, an unrestricted communication between two users requires the purchase of a communication token (Event 6) by one of them.

We now consider the characteristics of the user model time span and the consequences for the role of implicit models. In Table 2, Row [e], we observed that in reciprocal recommenders, many users leave the system after a successful recommendation. This is in contrast to classic recommenders where a really effective item recommender site should motivate longer term use. This means that in reciprocal recommenders, it is less common for users to have rich implicit preferences while they use the system (Row [f]). This is evident in the substantial use of online dating to find a long term partner, which aggravates the cold-start problem. This makes explicit user profiles and explicit preferences particularly valuable.

We analysed how long people used our online dating site. One quarter of users leave the website within four weeks of signing up. Of those who leave, half have had a paid communication within five days of their last login. Because paid communications can lead to a date (while unpaid communications cannot), it is possible that these users found a date before leaving the website. Strikingly, for those involved in a paid communication, over 10% used the website for less than one week. Despite their very short website use, they may have found a date.

The cold-start problem is very important in this domain. This is partly due to the short website use. In addition, the system must respond to the particular needs of reactive users, stereotypically the women who sign up and wait to receive EOIs (Table 3 Row [i]). The cold-start problem is well recognised for recommenders in general, and earlier work has been summarised by Schein et al. (2002). Recent ideas on the problem include integrating the rating of items into the sign-up process to increase engagement (Freyne et al., 2009), and using hybrid approaches such as performing linear regression on all user-item attribute pairs (Park and Chu, 2009). Akehurst et al. (2011) address the cold-start problem for an online dating recommender by finding users who have similar profiles to the cold-start user and providing recommendations based on the previous EOIs sent and replied by these users. This approach reduces the reliance on the user's explicitly stated preferences in terms of the attributes of objects.

In Section 3, we pointed to the distinctive character of reciprocal recommenders in having rich user profiles (Table 2 Row [a] and Row [c]). We argued that these may have inaccurate elements, both in the user profiles (Row [b]) and preferences (Row [d]). To gain understanding of this, we analysed how well the subject's explicit preferences match the object's profile. Figure 3 shows the percentage of EOIs where the subject's explicit preferences match and do not match the object's user profile, for the subjects who specified their explicit preferences. We can see that the largest mismatch is 30% for the attributes *education* and *religion* and the smallest is 10% for *age*, *smoke* and *height*. For example, 70% of the EOIs from subjects who specified preferred level of education were sent to objects who matched this preference and 30% were sent to objects who did not match it. This might be explained in two ways: subjects may have made an error in their preference profile or subjects may have found objects appealing in spite of their poor match on some attributes.

The blue line shows how many EOI were sent by users who specified their explicit preferences, as a percentage of all EOI. For instance, 14% of all EOIs were sent by subjects who specified preferred level of education. We can see that while almost all users have specified preferred age and 40-60% specified body type, smoking and height preferences, only 5% specified religion preferences and almost no one specified star sign preferences. This confirms, as expected, that the some attributes are more important than others when specifying user preferences.

We also investigated the effect of the explicit and implicit user preferences on the success rate (Akehurst et al., 2012). Using a slightly different but equally representative dataset (8,000 users with 116,000 interactions) we found that the explicit preferences are not good

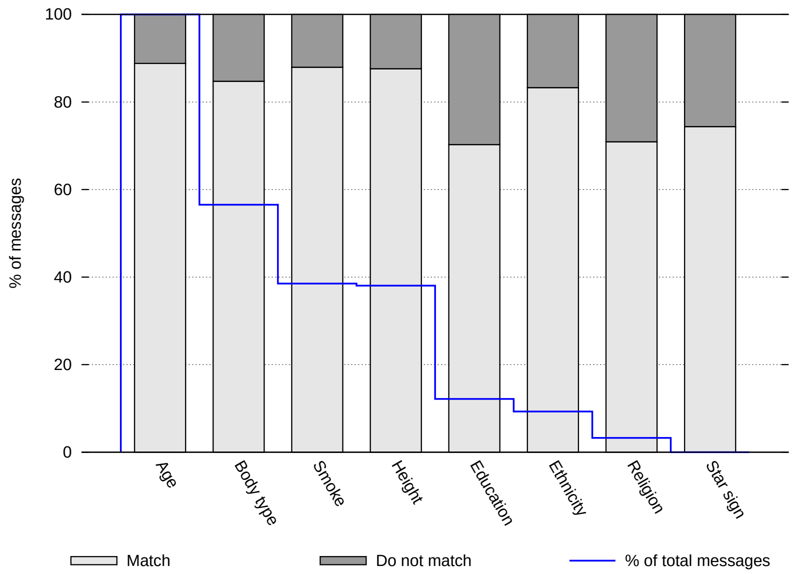

Fig. 3 Proportion of EOIs where the object matches the subject's explicit preference model

predictors of success rate. In particular, we found that in 62% of all successful interactions, the subject's explicit preferences did not match the object's profile and that in 42% of all unsuccessful interactions the subject's explicit preferences matched the object's profile. We proposed to learn the implicit user preferences from both successful and unsuccessful user interactions using a probabilistic machine learning approach. The results showed that the implicit preferences are good predictors of the success of user interactions achieving an accuracy of 89%. It is clear that we should exploit implicit preferences when they become available.

Despite the unreliability of the user input (Amatriain et al., 2009a,b), explicit preferences have an important place. They can help for new users when there is little data about implicit preferences. In Pizzato et al. (2010a), we found that explicit preferences were valuable for new users. We explored how many EOIs were needed to make the implicit profile useful. With just four EOIs, we were able to build implicit preferences that outperformed the explicit profile, based on the preferences provided by the users. This confirms the claims in Table 2.

5.4 User activity

Users of reciprocal recommenders may behave either proactively or reactively (Row [i] of Table 3.) In our introductory scenario of Section 1, we took the perspective of a proactive user (Bob), in the steps to marrying Alice. We now take the reactive perspective (of Alice):

1. Reactive user Alice signs up to the website and creates her profile;
2. Alice receives an EOI from Bob;
3. Alice views Bob's profile;
4. Alice replies positively to Bob;
5. As Bob purchases a token to communicate with Alice, she starts exchanging unconstrained messages with Bob and their relationship is taken offline;

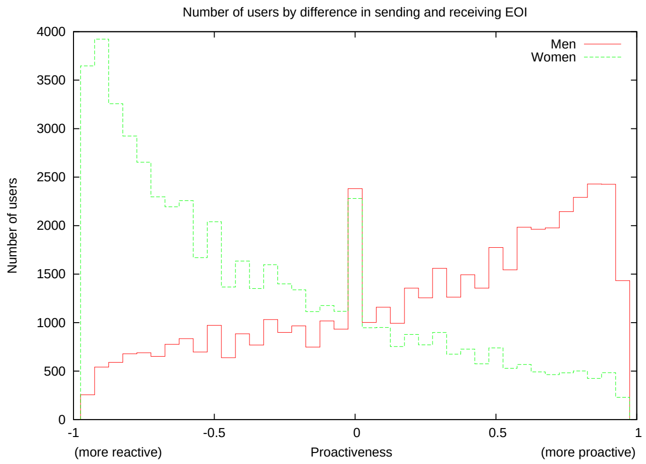

Fig. 4 Proactiveness by gender

6. Alice and Bob get married and they live happy ever after.

These two distinct classes of users approach the dating website differently and they have different needs of the recommendation. For instance, proactive users benefit most when good recommendations are *given* to them, while reactive users benefit when they are *recommended to* other users. We define the proactiveness of a user u as follows:

$$proactiveness(u) = \frac{(\#\text{EOIs sent} - \#\text{EOIs received})}{(\#\text{EOIs sent} + \#\text{EOIs received})} \tag{6}$$

Figure 4 shows the user proactiveness, for men and women separately. We can draw two important conclusions. First, for the whole user population, there is an equal balance of proactiveness and reactiveness. Second, in line with stereotypes of behaviour previously mentioned, the actual evidence indicates that more women are distinctively reactive, and more men are distinctively proactive.

We investigated how the recommender was affected by this aspect. We defined sized classes of equal numbers of users, according to their level of proactiveness or reactiveness. Table 7 shows these classes and the proportion of users by gender for each. (The highly reactive class, HR, is 81.7% female.)

Table 7 Classes of users according to how proactive or reactive they are

Abbr.	Class name	% of users (male/female)	Description
HR	Highly reactive	20% (18.3%/81.7%)	proactiveness $<$ -0.66
MR	Mildly reactive	20% (35.1%/64.9%)	proactiveness between -0.66 and -0.26
ERP	Equally proactive/reactive	20% (47.9%/52.1%)	proactiveness between -0.26 and 0.18
MP	Mildly proactive	20% (62.1%/37.9%)	proactiveness between 0.18, and 0.61
HP	Highly proactive	20% (76.3%/23.7%)	proactiveness $>$ 0.61

Table 8 Success rate, in terms of positive response to an EOI, according to the level of proactiveness and reactiveness of the subject and the object of EOIs (darker cells represent higher values)

Subject	Object HR	MR	ERP	MP	HP
HR	24.9%	30.1%	33.6%	38.9%	43.7%
MR	21.7%	26.1%	29.2%	34.2%	40.4%
ERP	18.0%	23.9%	26.4%	30.5%	36.6%
MP	14.6%	21.4%	24.9%	28.4%	34.5%
HP	9.0%	14.6%	19.6%	25.6%	35.1%

Table 9 Classes of users according to their level of activity (sending EOIs)

Abbr.	**Activity class**	**Users**	**Description**
VA	Very active	12%	Sent 33 or more EOIs
AA	Average active	54%	Sent between 2 and 32 EOIs
IA	Inactive	34%	Sent at most 1 EOI

Table 10 Classes of users according to their level of popularity (receiving EOIs)

Abbr.	**Popularity class**	**Users**	**Description**
VP	Very popular	11%	Received 32 or more EOIs
AP	Average popular	69%	Received between 2 and 31 EOIs
UP	Unpopular	20%	Received at most 1 EOI

We measured the success rates for each of these classes (see Table 8). The highest success rate (43.7%) occurs when highly reactive subjects (row HR) send EOIs to highly proactive objects (column HP). Overall, it appears that in cases where people, predominantly women, tend to be reactive but when they take the subject role, and send messages to proactive users (predominantly men), there is higher success. Now consider the other end of the spectrum, when the highly proactive user of row HP (predominantly men) sends messages to the highly reactive users of column HR (women), the chance of success is low (9%). These findings may be explained in terms of selectivity and popularity. For example, a person who is normally reactive (HR) may be particularly selective in the choice of the EOIs they send; these may be more successful because the subject has carefully assessed the object's profile. Similarly, people who are not selective in sending EOIs may have lower success. For popular users, who do not need to send EOIs, when they do send messages, these may be successful because they are very attractive.

Going beyond user proactiveness, we studied two related user characteristics: activity level and popularity, and their effect on user interactions and success rate. We defined three classes of users based on their activity as shown in Table 9 and three classes of user popularity as shown in Table 10. We then formed nine groups of users by combining both activity level and popularity, e.g. VP+VA is one of the groups and it includes users who are both very popular and very active. Table 11 and Table 12 summarise the EOIs sent and success rate for the nine groups, respectively.

As one would expect, Table 11 shows that most messages were sent by highly active users and received by highly popular objects. Consider the case of the very active users in subject role, the last three data row in the table. Summing the totals for these (in the *All* column) shows that this group, with just 12% of users, sent 58.3% of all EOIs. Now, consider this group in their object role, the last three data columns; the VA users only received 10.2% of all EOIs. So they send 5.7 EOIs for each one they receive. For the 11% of very popular

Table 11 EOI sent between users with different popularity and activity levels as a percentage of the total number of EOI (darker cells represent higher values)

	Object									
Subject	IA+UP	IA+AP	IA+VP	AA+UP	AA+AP	AA+VP	VA+UP	VA+AP	VA+VP	ALL
IA+UP	00.0%	00.4%	00.2%	00.0%	00.5%	00.5%	00.0%	00.1%	00.1%	01.8%
IA+AP	00.1%	00.8%	00.5%	00.1%	01.3%	01.1%	00.0%	00.4%	00.3%	04.6%
IA+VP	00.0%	00.1%	00.0%	00.0%	00.1%	00.1%	00.0%	00.1%	00.1%	00.4%
AA+UP	00.1%	01.3%	01.1%	00.1%	01.8%	02.2%	00.0%	00.3%	00.4%	07.1%
AA+AP	00.3%	04.2%	02.9%	00.5%	06.9%	06.1%	00.0%	01.7%	01.5%	24.1%
AA+VP	00.0%	00.5%	00.3%	00.1%	01.1%	00.7%	00.0%	00.5%	00.4%	03.7%
VA+UP	00.0%	00.8%	00.7%	00.0%	00.9%	01.6%	00.0%	00.1%	00.2%	04.3%
VA+AP	00.5%	08.1%	06.9%	00.7%	11.0%	14.7%	00.0%	01.0%	02.0%	44.9%
VA+VP	00.1%	01.6%	01.2%	00.2%	02.5%	02.4%	00.0%	00.5%	00.5%	09.1%
ALL	01.2%	17.6%	13.7%	01.8%	26.1%	29.3%	00.1%	04.7%	05.4%	100.0%

Table 12 Success rate of users with different popularity and activity levels (darker cells represent higher values)

	Object									
Subject	IA+UP	IA+AP	IA+VP	AA+UP	AA+AP	AA+VP	VA+UP	VA+AP	VA+VP	ALL
IA+UP	19.4%	13.7%	07.4%	14.7%	13.8%	07.7%	25.0%	25.9%	19.9%	12.5%
IA+AP	26.2%	21.8%	12.5%	28.9%	22.4%	12.9%	36.8%	31.1%	22.7%	20.0%
IA+VP	30.8%	35.4%	26.2%	41.6%	29.0%	23.4%	40.0%	35.4%	35.5%	30.9%
AA+UP	18.5%	14.2%	07.3%	20.9%	14.7%	07.6%	37.5%	23.2%	12.9%	11.7%
AA+AP	25.3%	20.2%	10.4%	26.1%	20.5%	11.6%	25.5%	28.5%	20.1%	17.8%
AA+VP	36.4%	31.9%	21.3%	33.0%	29.0%	20.5%	41.8%	36.1%	30.5%	28.7%
VA+UP	10.4%	08.7%	03.4%	11.2%	08.6%	03.9%	00.0%	15.7%	07.7%	06.1%
VA+AP	20.2%	14.1%	06.8%	18.4%	14.6%	07.7%	27.4%	24.3%	13.5%	11.4%
VA+VP	27.5%	20.7%	10.8%	26.0%	20.6%	12.2%	34.5%	31.3%	21.6%	18.0%
ALL	22.8%	16.9%	08.3%	23.0%	17.6%	09.2%	30.4%	28.6%	18.0%	14.4%

objects VP, the totals for columns 3, 6 and 9 show that in their roles as objects, they receive 48.4% of all EOIs and in subject role, they send 13.2%. So they receive 3.7 EOIs for each one they send. For the 54% of average activity users, AA, in subject role (rows 4-6) they sent 34.9% of all EOIs and in object role (columns 4-6) received 57.2%. So they received 1.6 EOIs for each one they sent.

We analysed the *success rate* of users in terms of their levels of activity and popularity. We summarise this in Table 12. The average success rate, in the lower right cell is 14.4%. Comparing each set of three rows, we can see that the third row (for VP subjects) is highest; for example, for the very inactive users (*IA*) in the first three data rows, it is 30.9%, compared with 12.5% and 20.0% for the others. This effect holds for all three activity levels. It means that very popular subjects were more likely to receive a positive response when they sent an EOI. (Note that as this table shows success rates, we can ignore size of each group and compare the numbers directly.) Now consider the effect of popularity. Comparing the total for very popular users in data rows 3, 6, and 9, we see the highest success rate for the least active IA subjects (data row 3) with 30.9% of EOIs. A similar trend applies for the other activity groups. This may also be explained in terms of selectivity: very inactive subjects (the first three data rows, IA) may well be more selective when they do send an EOI.

We can also observe in Table 12 that an object is less likely to respond positively to an EOI when they are popular. Unsurprisingly, the lowest success rate occurs for very active and unpopular subjects (VA + UP in row 7) where this row has the lowest success rate for each column. The best success rate occurs on the rare opposite occasions, when a very popular subject (rows 3, 6, 9) sends an EOI to a very active and unpopular object (VA + UP in column 7), with a maximum of 41.8% success rate.

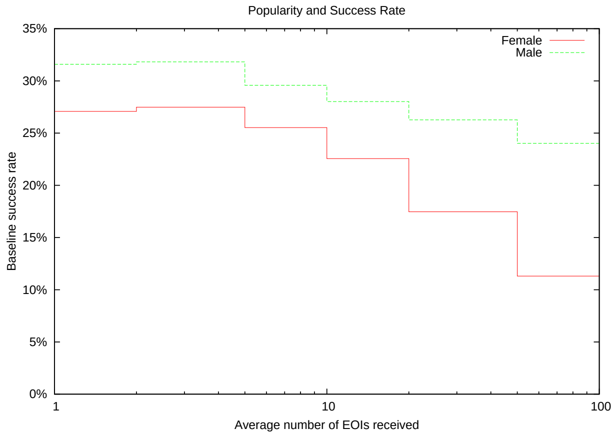

Fig. 5 Success rate as a function of receiver's popularity (average number of EOI received)

5.5 Creating recommenders that take account of user popularity

Row [m] of Table 4 note that the same object must not be recommended to too many others. This relates to the popularity and associated available of the potential object of the recommendation. In online dating, people can feel overwhelmed if they receive too many EOIs. Although a user may feel very flattered by receiving many EOIs, it may pose problems for them. These users may become overwhelmed with contact and because within this scenario they are unlikely to be able to consider all candidates presented, they may even leave the website. This problem is also important for other reciprocal recommender domains, such as employment websites where jobs that attract a very large number of applicants may cause information overload for the employer. Reciprocal recommenders can reduce the burden on popular users as they consider the preferences of both sides of the recommendation.

Figure 5 shows the relationship between the receiver's popularity, measured as the number of EOIs received, and the success rate of these EOIs. We can observe that users with high popularity are less likely to reply positively. Therefore, we can infer that popular users should not be recommended too often to other users. For instance, the success rate for a female user is 28% when she has received 4 EOIs and only 11.31% when she has received 50 or more EOIs. This indicates that, as expected, popular users pay less attention to each EOI (and its sender) than unpopular users do. However, we also found that users who received only one EOI have slightly lower success rate than users who have received between 2 and 5 EOIs.

Note that in Figure 5 we used data for users who have sent at least one EOI and that the success rate was computed by only considering the read EOI and ignoring the unread ones. The reason for this is to filter out the users who were never active. For instance, users may have signed up for the website but never used it; after a few hours or days, they might receive some EOIs. Because they are slowly fading out of the active users space, they will likely receive only a few messages; nevertheless, these unread messages can impact the success rate results for users with low popularity.

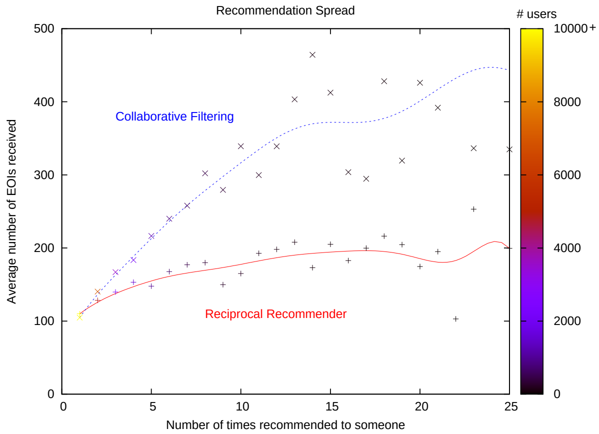

Fig. 6 The popularity of users (number of EOIs received) and the number of times they have been recommended to other users using a collaborative filtering and a reciprocal recommender approach

The effect of the popularity on the success rate is also clear from Table 12. We can see that the success rate decreases when unpopular subjects communicate with very popular objects and that the success rate is higher when both users have roughly the same popularity level.

The traditional collaborative filtering algorithms tend to recommend popular items because these items are liked by most users. However, reciprocal algorithms cannot do this. This is both because of the need to avoid overloading popular users and because they should account for the preferences of the object of the recommendation. That is, a user Bob might like the popular user Alice, but this does not mean that Alice will reciprocate.

We explored how our reciprocal recommender algorithms cope with these problems. In Figure 6 we compare our content-based reciprocal approach – RECON (Pizzato et al., 2010b) – with a standard collaborative filtering approach (user-based and non-reciprocal) in terms of recommending people with different popularity. The y-axis shows the popularity level of the user and the x-axis shows the average number of times an object with a given popularity level appeared in the top-10 recommendation list of subjects with average popularity and average activity. By selecting average users, we are sure that the subject's preferences are not weakly defined and also that the object's preferences are not biased against the subject. We can observe that the reciprocal recommender distributes the recommendations more evenly across all different levels of popularity (for the object) than the collaborative filtering recommender, which favours recommending popular objects.

Row [n] of Table 4 discusses the neglected user problem, which is the opposite of the overwhelmed popular user problem. Users who are never recommended to other users, or do not fit people's preferences are more likely to leave the website dissatisfied. Therefore, it is important that this group of users receive recommendations and that they are also recommended to people with whom they are likely to establish a successful communication.

The need to spread recommendations is a known problem for collaborative filtering approaches. Various methods have been used to relate similar items or users, in order to find new associations between them. For example, clustering on seldom selected items (Park and

Tuzhilin, 2008) has been suggested to find groups of similar items, demographic information has been used as another measure of similarity between people (Pazzani, 1999) and dimensionality reduction (Sarwar et al., 2000) was used to reduce the number of classes of items. In Pizzato and Silvestrini (2011a), we balanced the number of recommendations given to users by a collaborative filtering algorithm by applying stochastic matching. Our goal was to ensure that users receive the same number of recommendations as the number of times they are recommended to other users.

In summary, reciprocal recommenders are capable of evenly distributing users with various popularity levels in the recommendation list, and are therefore a good choice when this property is required. If there is a need to favour some groups of users over others that would not normally appear in recommendations, e.g. favouring unpopular users since they are more likely to respond positively (see Table 12), the preferences of these users can be given a higher weight as discussed in the next section.

5.6 Higher priority for a recommendation side or user group

Reciprocal recommendations account for the preferences of both parties; however, in some cases, the preferences of one party needs to be given higher priority than the other. This can be done by using a weighting technique to compute the recommendation, where higher weights are given to the preferences of the side with higher priority. A related idea, the use of weighed hybrid recommender, was discussed in Burke (2002). A weighted recommender assumes that there are several recommenders, each producing a score for a recommendation item, and its goal is to combine their scores in a final score. In the reciprocal case our goal is to combine the decisions of the recommenders for each side.

A weighting strategy is important for reciprocal recommenders since it allows the recommendation to match some business strategy such as giving higher priority to different classes of users. For instance, if a job recommendation charges companies to post jobs and is free for job candidates, it is reasonable to give higher importance to the preferences of companies. Similarly, if an online dating website provides a free and a subscription model for users, subscribers can be given higher priority when generating the recommendations.

In the recommendation algorithm presented in (Pizzato et al., 2010b), we can easily account for the different priorities of the recommendation sides and user groups, by introducing weights in the compatibility score which reflect the the role of the user (subject or object) and its group. Equation 7 shows an example of weighted reciprocal compatibility score s_y for a user x. The weight W_u represents a specific weight for one particular user u or their class; the weigh ω_s represents the importance of the subject's preferences and the weight ω_o represents the importance of the object's preferences.[4]

$$s_y \leftarrow \frac{(W_x \times \omega_s) + (W_y \times \omega_o)}{\left(\frac{W_x \times \omega_s}{Compatibility(P_x, y)}\right) + \left(\frac{W_y \times \omega_o}{Compatibility(P_y, x)}\right)} \tag{7}$$

The use of a reciprocal compatibility score can also help to make recommendations for users who do not have preference information. For instance, when the subject's preferences are unreliable (e.g. new users), we may choose to set ω_s to 0, therefore relying entirely on the object's preference for recommendations. It is also possible to give higher weights to

[4] Note that ω_s and ω_o are between 0 and 1 and that $\omega_s = (1 - \omega_o)$.

different user classes; for instance, if a user x is a subscriber, we may set its weight W_x to a higher value than the weights of the other users who do not hold a subscription.

Weights can also be assigned according to other known information about the users. For instance, since we know that popular users are less likely to respond positively, it is reasonable to assign weights to the users' preference models that are inversely proportional to the users' popularity. For instance, given two users Alice who equally match user Bob's preferences and whose preferences equally match Bob's profile, then the user Alice who has the lower popularity will rank higher in Bob's recommendations.

5.7 Reducing the risk of people being rejected

Bad recommendations can have a strong negative impact on reciprocal recommender systems, as highlighted in Table 4, rows [k] and [l]. There are two main types of bad recommendations: recommendations that do not match the user preferences and recommendations that are not likely to be reciprocated. If users receive recommendations that do not suit their preferences, they will distrust the system and its future recommendations. If the users like the recommendations but are frequently rejected by the recommended people, they will feel disappointed, lose confidence and leave the website.

As the cost of rejection is very high, we should aim at reducing the risk for it by using both positive and negative user preferences. One method for doing this is to build a classifier trained on positive and negative examples and then use it to recommend people who are confidently classified as positives.

Another method is to rank the list of recommendations using a criterion that considers how much two people like and dislike each other. Our compatibility score (see Algorithm 2) is based on the predicted user likes; we can define a similar score for the user dislikes and combine the two into an overall compatibility score computed as the difference between the positive and negative scores, normalised between 0 and 1:

$$C^{\pm}(A,B) = \frac{1 + C^{+}(A,B) - C^{-}(A,B)}{2} \tag{8}$$

In Equation (8) from Pizzato et al. (2011) $C^{+}(A,B)$ represents the positive compatibility score between the user A and a user B, given the EOIs sent by A (i.e. using A's likes); $C^{-}(A,B)$ represents the negative compatibility between the same users given A's negative responses to EOIs that he/she has received.

In a nutshell, the positive compatibility score between A and B measures the similarity between B and the people A likes, while the negative compatibility score between A and B measures the similarity between B and the people A dislikes. The overall compatibility score between A and B measures how much B matches the people A likes and dislikes; if it is greater than 0.5, B is more similar to the people A likes, if it is smaller than 0.5, B is more similar to the people A dislikes. Thus values close to 0 indicate negative preference and values close to 1 indicate positive preference.

Our results showed that all three compatibility scores were normally distributed across the number of EOIs sent. The mean value and standard deviations were as follows: 0.45 +/- 0.07 for the positive compatibility score, 0.40 +/- 0.09 for the negative compatibility score and 0.51 +/- 0.08 for the overall compatibility score. These results show that on average the positive compatibility score is greater than the negative compatibility score between senders and receivers.

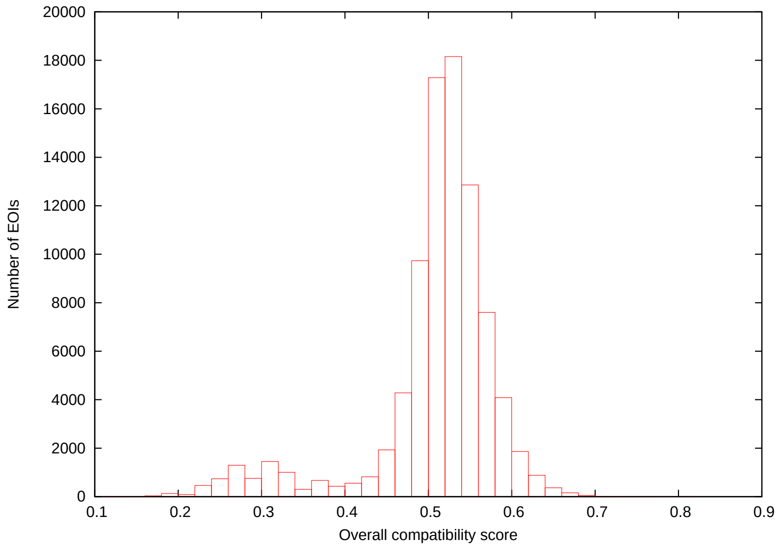

Fig. 7 Overall compatibility score as a function of the EOI sent

Figure 7 shows the results for the overall compatibility score. Overall, 72% of the EOIs were sent to an object with an overall compatibility score higher than 0.5. This means that users are more likely to send EOIs to people who are more similar to the people they like than to the people they dislike.

Unlike other recommendations, reciprocal recommendations benefit from negative preferences as can be seen in Figure 8. This occurs because negative preferences are modelled using negative *responses*, therefore improving measures that account for the response of the users. We can observe that the use of negative preferences in the reciprocal recommender gives a better success rate for top-1 and for top-5, while for top-10 and for top-100 they are virtually the same. The results for the reciprocal combined recommender for top-1 and for top-5 are 37.46% and 30.77% respectively, while the corresponding results for the reciprocal positive-only recommender are 31.78% and 28.09%. We ran the Mann-Whitney-Wilcoxon test on both data sets, for different values of n, to see whether the success rate improvement of the negative preference reciprocal recommender was statistically significant. We found that the difference is significant to a 95% confidence interval from top-1 to top-5 but is not significant at higher n. Importantly, the success rates of both recommenders are higher than the baseline success rate, which is the ratio between the number of positively replied EOIs and the number of EOIs in the test set. These results are important for our domain, particularly for the case of unpopular users, for whom we may have small numbers of good recommendations. These results are also important as the very top recommendations are critical because people are most likely to focus time and attention on the first set of items presented to them (Joachims et al., 2005).

Figure 9 shows the failure rate results. There is a lower failure rate ($F@n$) for reciprocal preferences when negative preferences are used, compared to the case when only positive

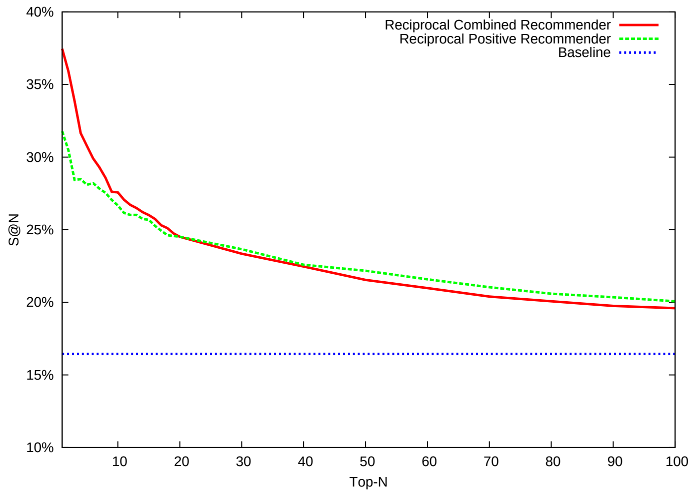

Fig. 8 Success rate for different number of recommendations given

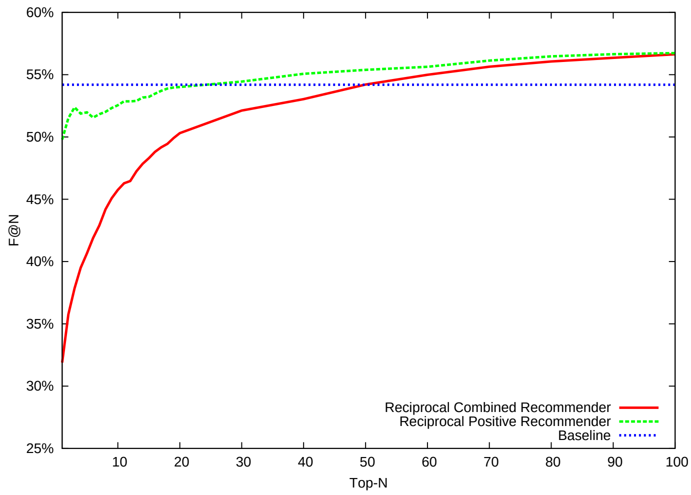

Fig. 9 Failure rate for different number of recommendations given

preferences are used. We can observe that for all values of n the combined positive and negative recommender consistently outperforms the positive only recommender. But at top-100, the failure rate increases above the baseline level of 54.19%, which indicates that the reciprocal recommender provides recommendations with lower chance of rejection only for lower values of n.

Overall these results show that exploiting negative preferences is a promising approach for fine tuning reciprocal recommendations. As the negative preference compatibility score is subtracted from the positive one, the effect is in fact only pushing recommended users down the list, not promoting users to be recommended who did not already have a very high positive score.

6 Other recommendation aspects

In this section we discuss two other important aspects of recommender systems - serendipity and transparency. They are also differentiating features between the traditional and reciprocal recommender systems. For instance, in traditional recommender systems, serendipity is normally viewed as beneficial for the user receiving recommendations. In reciprocal recommenders, its benefit is unclear. Similarly, in traditional recommender systems the transparency of the recommender is a highly desirable feature but in reciprocal recommenders it may cause some privacy problems.

6.1 Serendipity

It is unclear how much serendipity can be allowed in a reciprocal domain, and how it should be presented. Indeed, for a system to allow serendipity it must be able to recommend objects outside the person's own explicit preferences. In a traditional domain, especially where people have not invested a lot of effort providing their detailed preferences, serendipity is more acceptable. However, as we highlighted in Table 2, users in reciprocal domains generally spend time filling out their profile and explicit preferences. Therefore serendipitous recommendations that disregard this information may be unacceptable as it can cause users to distrust the system. As discussed in Section 5.7, bad recommendations can have a strong negative impact in reciprocal domains.

On the other hand, allowing some level of surprise in the recommendation has been shown to be an advantage for recommender systems (Sinha and Swearingen, 2001). Finding surprising matches might be a desired feature for some online dating users who might like to find someone different from the people they normally meet in person or from the object they thought they were seeking.

A possible way to allow for serendipity, without the risks noted above, is to explicitly inform users when recommendations deviate from their explicit preferences and even to allow them to choose the serendipity level.

6.2 Transparency

The transparency of a recommender, or the degree to which the user can understand why a particular recommendation is made, has been shown to be an important feature of recommender systems. It gives users a sense of security, confidence and trust in the system. As shown by Sinha and Swearingen (2002), providing an explanation together with the recommendation, increases the user trust in the recommender system. However, recommendation transparency for reciprocal recommenders introduces new challenges. In traditional collaborative filtering recommenders, recommendations are generated based on the preferences of the user and those of a set of anonymous similar users. and it is easier to present an explanation without violating the privacy of the other people. However, in reciprocal recommenders providing such explanation is more difficult as it requires revealing the preferences of the object of the recommendation. This is not always appropriate and may also raise some privacy issues. For instance, it is ok to tell Bob that he was recommended Alice because he liked another user named Clare; however, it is problematic to tell Bob that Alice was recommended to him because she liked another user named Daniel.

7 Discussion of broader reciprocal recommender domains

The empirical work in this paper has been in the context of online dating. However, our exploration of the nature of reciprocal recommenders identified characteristics that are relevant for a far broader class of recommender systems. These include the many domains which involve reciprocal relationships between people. We now discuss some of these, noting the nature of the reciprocity. For all these domains, our work on reciprocal recommenders has the potential to be valuable in taking account of the characteristics of reciprocal recommender domains.

7.1 Friends and acquaintances search

Finding a known friend, or establishing a new friendship requires reciprocity. Social networks such as Facebook[5] and LinkedIn[6], require people to reciprocate friend relationships. People can invite others to become their “friend”, but the connection is only established when both people acknowledge the relationship.

Despite clear similarities between finding a friend and finding a date, there are important differences. For instance, dating relationships are strictly private; by contrast, friend relationships in social networks are normally visible to other users (with some degree of privacy control by the user). This has clear implications for the friend finding problem, since public friend information can be used to find friends of friends who might also be someone else’s friend.

Patil (2009) observed the connections in a social network and found a higher similarity of interests among friends. Sites such as Twitter and Flickr[7] encourage users to establish contacts based on interests (i.e. same taste in photography), and not necessarily on an existing relationship in real life. Because of this, these sites have similarities with online dating. However, Patil (2009) also found that similar interest is not the motivating factor for people in speed dating to meet. Similarly, sites such as Flickr have successfully used the friend-of-a-friend relationship to capture the user interest and recommend new connections. In online dating, it is mostly untrue that a date of a date can be someone else’s date.

7.2 Job recommender

Job recommendations share many attributes with online dating. One can see a career relationship between employer and employee as a long term and stable relationship. Similarities

[5] `http://facebook.com`
[6] `http://linkedin.com`
[7] `http://flickr.com`

apart, the employment relationship does not share the equality expected from a romantic relationship. Employees as individuals are, on average, in a lower and more submissive position than employers. Obviously, for certain areas the demand for highly specialised positions causes employees to have higher bargaining power.

This balance of bargaining power is very important for a reciprocal recommender, since it may be necessary to adjust whose preferences are more important to be matched. For instance, if the number of suitable candidates for a job position is too high, it might be desirable to give higher importance to the employer's preferences instead of the candidate's preferences. Similarly, if there are many job positions and few candidates then the candidate preferences may be given higher priority. A weighted approach should also consider the vested interest of both sides of the recommendation. For instance, if the employment website is free for candidates but paid by employers, then a weighted recommended approach might give higher weights to the employer.

A weighted strategy can ensure a balanced strategy that can prevent spamming, which is detrimental to both sides of the recommendations process. Spamming in the job seeking process can occur if the candidates apply indiscriminately for many positions, even if they lack the required skills. Even more importantly, spamming can happen when employers or job agency companies start contacting candidates for positions that they are not interested in.

Other specialised areas of job recommendations such as finding a cleaner, a baby sitter, a musician, or a band member are particularly reciprocal. These specialised areas can also involve added dimensions such as trust and reputation of both parties involved. For instance, both baby sitter and families can have different levels of commendation from those they have hired or worked for in the past.

Reputation is relevant for a normal job recommender, where it is based on various factors, such as recommendations of previous employers and reviews from previous and current employees. Systems such as LinkedIn, which implement a professional social network where people post their professional contacts and their curriculum vitae as profiles, also allow people to write recommendation letters for other users.

Outsourcing services such as those provided by websites such as Freelancer[8] allow small companies to outsource small jobs to people with the required skill set. This is another domain that greatly benefits from reciprocity and reputation.

7.3 Other domains

There are many other domains that involve people-to-people recommenders, with the need for reciprocity. We illustrate the character of just a few of these.

Finding flatmates. Flatmate searching is another example of a reciprocal domain, where one party is searching for someone to rent a bed or a room and the other party is searching for a bed or room to live in. Both parties have their own constraints and requirements to be satisfied before a successful rental agreement can be established.

Matching students with tutors or mentors. A student might seek a mentor who knows the nominated subject and is willing to help them. A mentor or tutor might also be seeking students who want their help and guidance. There are different situations involving relationships between students and tutors. In some, such as a university classroom, pragmatic issues mean that there is little choice in the matching of tutors and students. However, there are

[8] `http://freelancer.com`

many cases where high quality reciprocal matching could be extremely valuable. This includes the range of cases involving one-on-one tutoring, mentoring or advising; for example for research supervision, trouble-shooting or remedial tutoring.

Finding experts and research collaborators. Finding people with appropriate expertise, or research collaborators in the academic community, is important for many tasks and especially for complex and highly specialised problems. This is similar to the matching of students with tutors; it depends on the skills of the expert and also on the preferences and motivation of both parties.

Selection reviewers for papers. When reviewing papers for conferences and journals, choice of reviewers is typically based on their expertise in the research topic presented in the paper. Other factors are the reviewer's availability, quality of previous reviews and lateness in submitting previous reviews. Some journals require that authors suggest a list of potential reviewers and also reviewers who should not be selected due to bias or conflict of interest. In these cases people to people recommenders can be applied to suggest potential reviewers. However, these recommenders are not necessarily reciprocal.

8 Summary and Conclusions

The initial motivation for our research was to create improved recommenders for a large online dating site and an employment site. As we analysed the nature of these tasks, the importance of reciprocity became apparent and we coined the term *reciprocal recommender*. Given the extensive body of recommender research, we were surprised to discover that there had been so little study of this class of recommender, even though there are many important contexts where these recommenders can play significant roles. Nor had there been any systematic analysis of the nature of such reciprocal recommenders, how they differed from traditional recommenders and how this impacts the design of their recommender algorithms.

This paper presents the result of our analysis of the nature of reciprocal recommenders. Our first attempt at this task (Pizzato et al., 2010b) identified some of their important characteristics. Since then, we have completed a range of experiments and studies, which highlight the importance of reciprocal recommenders and shows their distinctive nature.

Our characterisation of reciprocal recommenders was a starting point to define the research needed to understand ways of building effective reciprocal recommenders. This paper brings together results from several earlier papers and internal theses: Akehurst (2010); Pizzato et al. (2010b,c,a, 2011); Akehurst et al. (2011); Pizzato and Silvestrini (2011a); Akehurst et al. (2012). We also introduce new results that further explores the nature of reciprocal recommenders. Section 5 integrates these. Since that has included much detail, we now summarise it, in terms of the characteristics of reciprocal recommenders.

First, we tackled the very top level aspect: how to create a recommender that takes account of reciprocity and how much that impacts performance. Section 5.1, reports how the top-10 recommendations by our recommender system RECON (Pizzato et al., 2010b) achieves a success rate of 42%, a considerable improvement over the non-reciprocal recommender's success rate of 23%. Facing the cold start problem for new users, RECON also improves the results considerably.

For the creation of the user model, Table 2 identified a number of distinctive features of reciprocal recommenders, which were studied in Section 5.3. We have shown that users were more active when they had more complete profiles. Similarly, users with longer text profiles were more active and received more EOIs. This confirms the advantage of having a complete profile in online dating. In Section 5.3. we have observed that users often select characteristics in their explicit preferences that do not match the characteristics of people they seek communication with (their implicit preferences). For some characteristics, such as when users have shown a preference for *education* and *religion*, the mismatch between explicit and implicit preferences can be as high as 30% of all their communications. We have also summarised results of experiments in Akehurst et al. (2012), which shows that 62% of all successful interactions had mismatches between the subject's explicit preferences and the object's profile. On the flip side, in 42% of all unsuccessful interactions, the subject's explicit preferences did match the object's profile. Our study indicates that we can achieve a higher accuracy if we disregard the user's explicit preferences and use solely the user's implicit preferences. This has important implications because users may become disgruntled with recommendations that goes against their stated preferences. Since our work was based on historic data, future work needs to explore this.

In Table 3 we argued that in the reciprocal recommenders domain success is determined by both subject and object and that the subject is aware of this. We explored this in Section 5.2, finding that when subjects moved from browsing to a serious attempt to communicate with an object (i.e. actually paying for a communication), they were more likely to select those objects who explicitly sought people like them. In Section 5.3, we report that a significant proportion of users leave the website early after a possibly successful communication. This reinforces our argument that reciprocal recommenders must be able to perform well, even without a rich model of implicit preferences.

Our studies in Section 5.4 explored the asymmetry in user activity based on role. We confirmed the stereotype that women tend to be reactive and men proactive and that user activity and popularity are strongly related to success. Our analysis points to the perils of overloading popular reactive users and the need to account for very active users. Our active users sent 5.7 EOIs for each one they received, where popular users receive 3.7 EOIs for each one they send. A similar picture emerges for the success of the EOIs sent. The least active users have the highest success for the small number of EOIs they sent. Popular users are similarly successful.

The nature of people-to-people, reciprocal recommenders means that there are several success measures beyond those of traditional recommenders, taking the perspective of each party and the connection between them. One of these relates to poor recommendations, where these could have a very serious cost, especially if a person suffers repeated rejections. This combines with the user's need for just a small number of very good recommendations. In Section 5.7, we showed that a recommender built to model negative preferences (Pizzato et al., 2011) could improve the success rate of top-1 and top-5 recommendations whilst reducing the failure rate on the full range of top-*N* recommendations.

While we have reported extensive studies on a very large data sets, much remains to be done. In Section 6 we particularly noted the need to study serendipity and transparency. While our empirical work has been extensive, it has been limited to a single domain, online dating, and the historical data of one data set (albeit a very large one with 1.6 million users). In Section 7, we identified some of the many domains that involve recommending people to people and we discussed some of the aspects that are likely to differ across some of these. While the large data set has enabled us to explore a broad range of issues, deployment of a selection of recommenders will provide insights into the ways that these can alter behaviour and how user interface elements can be combined with recommender approaches. Another important direction for future research relates to the success measures. This paper has primarily assessed success in terms of the response to an EOI and top-*N* recommendations. This is a good choice for historic data studies but other measures for assessing live performance are very important. Another promising direction for future research is to create a toolkit of reciprocal recommender analysis and recommender algorithms.

Our goal in this paper was to provide a consolidated view of reciprocal recommender, drawing upon our studies on historic data from an online dating site. This can serve as a foundation for future work on the important class of recommender systems that aim to help match people to people. We look forward to similar analyses that explore other reciprocal recommender domains, to gain understanding of them and how they compare with our work in online dating. Similarly, our algorithms can provide a starting point for designing these recommenders. Our work has made us aware of many potential contexts where reciprocal recommenders can be valuable, such as matching employers and employees, flatmates, and tutors and learners. We hope that our characterisation of its nature and our studies will facilitate creation of recommenders in many people-to-people domains that demand reciprocity.

Acknowledgements This research was carried out as part of the activities of, and funded by, the Smart Services Cooperative Research Centre (CRC) through the Australian Government's CRC Programme (Department of Innovation, Industry, Science and Research).

References

M. Agrawal, M. Karimzadehgan, and C. Zhai. An online news recommender system for social networks. In *Proceedings of the Workshop on Search in Social Media (SSM 2009), co-located with ACM SIGIR 2009 Conference on Information Retrieval*, Boston, July 23 2009.

J. Akehurst. A hybrid content-collaborative reciprocal recommender. Honours thesis, School of Information Technologies, The University of Sydney, 2010.

J. Akehurst, I. Koprinska, K. Yacef, L. Pizzato, J. Kay, and T. Rej. CCR - a content-collaborative reciprocal recommender for online dating. In *Proceedings of the 22nd International International Joint Conference on Artificial Intelligence (IJCAI 2011)*, Barcelona, Spain., July 2011.

J. Akehurst, I. Koprinska, K. Yacef, L. Pizzato, J. Kay, and T. Rej. Explicit and implicit user preferences in online dating. In L. Cao, J. Huang, J. Bailey, Y. Koh, and J. Luo, editors, *New Frontiers in Applied Data Mining*, volume 7104 of *Lecture Notes in Computer Science*, pages 15–27. Springer Berlin / Heidelberg, 2012. ISBN 978-3-642-28319-2.

X. Amatriain, J. M. Pujol, and N. Oliver. I like it... i like it not: Evaluating user ratings noise in recommender systems. In *UMAP '09*, pages 247–258, Berlin, Heidelberg, 2009a. Springer-Verlag. ISBN 978-3-642-02246-3.

X. Amatriain, J. M. Pujol, N. Tintarev, and N. Oliver. Rate it again: increasing recommendation accuracy by user re-rating. In *RecSys '09: Proceedings of the third ACM conference on Recommender systems*, pages 173–180, New York, 2009b. ACM. ISBN 978-1-60558-435-5.

N. Bansal, A. Gupta, J. Li, J. Mestre, V. Nagarajan, and A. Rudra. When lp is the cure for your matching woes: Improved bounds for stochastic matchings. In M. de Berg and U. Meyer, editors, *Algorithms – ESA 2010*, volume 6347 of *Lecture Notes in Computer Science*, pages 218–229. Springer Berlin / Heidelberg, 2010.

T. Blecker, N. Abdelkafi, G. Kreutler, and G. Friedrich. An advisory system for customers' objective needs elicitation in mass customization. In *4th Workshop on Information Systems for Mass Customization (ISMC 2004) at the fourth International ICSC*

Symposium on Engineering of Intelligent Systems (EIS 2004), University of Madeira, Funchal/Portugal, February 28 - March 3 2004. S. 1 – 10.

L. Brožovský and V. Petříček. Recommender system for online dating service. *CoRR*, abs/cs/0703042, 2007.

S. Bull, J. E. Greer, G. I. McCalla, L. Kettel, and J. Bowes. User modelling in i-help: What, why, when and how. In *User Modeling*, volume 2109 of *Lecture Notes in Computer Science*, pages 117–126, 2001. ISBN 3-540-42325-7.

R. Burke. Hybrid recommender systems: Survey and experiments. *User Modeling and User-Adapted Interaction*, 12(4):331–370, 2002. ISSN 0924-1868. doi: 10.1023/A:1021240730564.

X. Cai, M. Bain, A. Krzywicki, W. Wobcke, Y. S. Kim, P. Compton, and A. Mahidadia. Learning collaborative filtering and its application to people to people recommendation in social networks. In *Proceedings of the 2010 IEEE International Conference on Data Mining*, ICDM '10, pages 743–748, Washington, DC, USA, 2010. IEEE Computer Society. ISBN 978-0-7695-4256-0.

X. Cai, M. Bain, A. Krzywicki, W. Wobcke, Y. S. Kim, P. Compton, and A. Mahidadia. Collaborative filtering for people to people recommendation in social networks. In *AI 2010: Advances in Artificial Intelligence*, volume 6464/2011 of *Lecture Notes in Computer Science*, pages 476–485. Springer-Verlag, 2011.

J. Chen, W. Geyer, C. Dugan, M. Muller, and I. Guy. Make new friends, but keep the old: recommending people on social networking sites. In *CHI '09*, pages 201–210, New York, 2009a. ACM. ISBN 978-1-60558-246-7.

N. Chen, N. Immorlica, A. R. Karlin, M. Mahdian, and A. Rudra. Approximating matches made in heaven. In *Proceedings of the 36th International Colloquium on Automata, Languages and Programming: Part I*, ICALP '09, pages 266–278, Berlin, Heidelberg, 2009b. Springer-Verlag. ISBN 978-3-642-02926-4.

F. Diaz, D. Metzler, and S. Amer-Yahia. Relevance and ranking in online dating systems. In *SIGIR '10: Proceeding of the 33rd international ACM SIGIR conference on Research and development in information retrieval*, pages 66–73, New York, NY, USA, 2010. ACM. ISBN 978-1-4503-0153-4.

M. Fazel-Zarandi, H. J. Devlin, Y. Huang, and N. Contractor. Expert recommendation based on social drivers, social network analysis, and semantic data representation. In *Proceedings of the 2nd International Workshop on Information Heterogeneity and Fusion in Recommender Systems*, HetRec '11, pages 41–48, New York, NY, USA, 2011. ACM. ISBN 978-1-4503-1027-7.

J. Freyne, M. Jacovi, I. Guy, and W. Geyer. Increasing engagement through early recommender intervention. In *RecSys '09: Proceedings of the third ACM conference on Recommender systems*, pages 85–92, New York, 2009. ACM. ISBN 978-1-60558-435-5.

D. Gale and L. S. Shapley. College admissions and the stability of marriage. *American Mathematical Monthly*, 69(1):9–15, 1962. ISSN 0002-9890.

M. Gelain, M. S. Pini, F. Rossi, K. B. Venable, and T. Walsh. Local search for stable marriage problems with ties and incomplete lists. *CoRR*, abs/1007.0637, 2010.

J. Greer, G. McCalla, J. Collins, V. Kumar, P. Meagher, and J. Vassileva. Supporting peer help and collaboration in distributed workplace environments. *International Journal of Artificial Intelligence in Education*, 9(1998):159–177, 1998.

A. Gunawardana and G. Shani. A survey of accuracy evaluation metrics of recommendation tasks. *The Journal of Machine Learning Research*, 10:2935–2962, 2009. ISSN 1532-4435.

T. Joachims, L. Granka, B. Pan, H. Hembrooke, and G. Gay. Accurately interpreting clickthrough data as implicit feedback. In *Proc. of the 28th annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '05, pages 154–161, New York, NY, USA, 2005. ACM. ISBN 1-59593-034-5.

Y. S. Kim, A. Mahidadia, P. Compton, X. Cai, M. Bain, A. Krzywicki, and W. Wobcke. People recommendation based on aggregated bidirectional intentions in social network site. In *Knowledge Management and Acquisition for Smart Systems and Services*, volume 6232/2010 of *Lecture Notes in Computer Science*, pages 247–260. Springer-Verlag, 2010.

S. Kutty, L. Chen, and R. Nayak. A people-to-people recommendation system using tensor space models. In D. Shin, editor, *ACM Service-Oriented Architectures and Programming: 27th Annual ACM Symposium on Applied Computing (SAC)*, Riva del Garda, Trento, 2012. ACM Press.

J. Malinowski, T. Keim, O. Wendt, and T. Weitzel. Matching people and jobs: A bilateral recommendation approach. In *Proceedings of the 39th Annual Hawaii International Conference on System Sciences*, volume 6, page 137c, 2006.

B. McFee and G. Lanckriet. Metric learning to rank. In *Proceedings of the 27th International Conference on Machine Learning (ICML'10)*, June 2010.

S. T. Park and W. Chu. Pairwise preference regression for cold-start recommendation. In *RecSys '09: Proceedings of the third ACM conference on Recommender systems*, pages 21–28. ACM, 2009. ISBN 978-1-60558-435-5.

Y.-J. Park and A. Tuzhilin. The long tail of recommender systems and how to leverage it. In *RecSys '08*, pages 11–18. ACM, 2008. ISBN 978-1-60558-093-7.

A. N. Patil. Homophily based link prediction in social networks. Technical report, Department of Computer Science, Stony Brook University, 2009. URL `http://www.cs.stonybrook.edu/~akshay/rpe.pdf`.

M. J. Pazzani. A framework for collaborative, content-based and demographic filtering. *Artif. Intell. Rev.*, 13(5-6):393–408, 1999. ISSN 0269-2821.

L. Pizzato, T. Chung, T. Rej, I. Koprinska, K. Yacef, and J. Kay. Learning user preference in online dating. In E. Hüllermeier and J. Fürnkranz, editors, *Proceedings of the Preference Learning (PL-10) Tutorial and Workshop, European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECML PKDD)*, September 2010a. URL `http://www.ke.tu-darmstadt.de/events/PL-10/papers/8-Pizzato.pdf`.

L. Pizzato, T. Rej, T. Chung, I. Koprinska, and J. Kay. RECON: a reciprocal recommender for online dating. In *RecSys '10: Proceedings of the fourth ACM conference on Recommender systems*, pages 207–214, New York, NY, USA, 2010b. ACM. ISBN 978-1-60558-906-0.

L. Pizzato, T. Rej, T. Chung, K. Yacef, I. Koprinska, and J. Kay. Reciprocal recommenders. In *8th Workshop on Intelligent Techniques for Web Personalization and Recommender Systems, UMAP'2010*, Hawaii, USA, 20-24 June 2010c.

L. Pizzato, T. Rej, K. Yacef, I. Koprinska, and J. Kay. Finding someone you will like and who won't reject you. In *Proceedings of the 19th International Conference on User Modeling, Adaptation and Personalization (UMAP 2011)*, Girona, Spain, July 2011.

L. A. Pizzato and C. Silvestrini. Stochastic matching and collaborative filtering to recommend people to people. In *Proceedings of the fifth ACM conference on Recommender systems*, RecSys '11, pages 341–344, New York, NY, USA, 2011a. ACM. ISBN 978-1-4503-0683-6.

L. A. Pizzato and C. Silvestrini. Stochastic matching and collaborative filtering to recommend people to people. Technical Report 676, School of Information Technologies, The

University of Sydney, 2011b.
Reuters. Time constraints boost popularity of online dating. Reuters Life!, 25 January 2010. URL `http://www.reuters.com/article/idUSTRE6004PC20100125`. Accessed on the 20th of October 2010.
D. Richards, M. Taylor, and P. Busch. Expertise recommendation: A two-way knowledge communication channel. *Autonomic and Autonomous Systems, International Conference on*, 0:35–40, 2008.
E. Ronn. Np-complete stable matching problems. *Journal of Algorithms*, 11(2):285 – 304, 1990. ISSN 0196-6774.
B. M. Sarwar, G. Karypis, J. A. Konstan, and J. T. Riedl. Applications of dimensionality reduction in recommender systems – a case study. In *WebKDD Workshop at the ACM SIGKKD*. ACM, 2000.
A. I. Schein, A. Popescul, L. H. Ungar, and D. M. Pennock. Methods and metrics for cold-start recommendations. In *SIGIR '02*, pages 253–260, 2002. ISBN 1-58113-561-0.
R. Sinha and K. Swearingen. The role of transparency in recommender systems. In *CHI '02 extended abstracts on Human factors in computing systems*, CHI '02, pages 830–831, New York, NY, USA, 2002. ACM. ISBN 1-58113-454-1.
R. R. Sinha and K. Swearingen. Comparing recommendations made by online systems and friends. In *DELOS Workshop: Personalisation and Recommender Systems in Digital Libraries*, 2001.
L. Terveen and D. W. McDonald. Social matching: A framework and research agenda. *ACM Transactions on Computer-Human Interaction*, 12(3):401–434, 2005.

Author Biographies

Luiz Augusto Pizzato is a Research Fellow at the University of Sydney working for the Smart Services Corporate Research Center. Luiz completed his PhD in 2009 at Macquarie University, Australia in the area of natural language processing. His primary interest lie in the areas of recommender systems, information retrieval, natural language processing, social networks and fraud detection.

Tomasz Rej is a software developer currently working at Roamz, the company's product being a mobile phone app that helps you find interesting places around you. Tomasz received a B.CS. (Honours) from the University of Sydney in 2009, and after that he joined the University of Sydney as a research assistant for the Smart Services CRC. Tomasz is also a Chess FIDE Master.

Joshua Akehurst is currently completing a M.Phil at the University of Sydney working on personalisation and recommender systems. Joshua completed his B.CS. (Honours) from the University of Sydney in 2011, and has worked as a research assistant for the Smart Services CRC. His main research interests are: machine learning, recommender systems, natural language processing and data mining.

Irena Koprinska is a Senior Lecturer at the University of Sydney. She received her PhD in Computer Science from the Institute for Information Technologies in Sofia and Masters in Higher Education from the University of Sydney. Irena was a visiting researcher at the

Technical University of Graz, Austria, and a post-doctoral fellow at the University of Trieste, Italy and the University of Otago, New Zealand. Her research interests are in Machine Learning, Data Mining and Neural Networks, and their applications for Pattern Recognition, Recommender Systems and Personalisation. She regularly serves on the program committees of international conferences and also as a reviewer for journals and funding bodies.

Kalina Yacef is a Senior Lecturer in the Computer Human Adaptive Interaction (CHAI) research lab, at the University of Sydney. She received her PhD in Computer Science from University of Paris V in 1999. Her research interests spans across the fields of Artificial Intelligence and Data Mining, Personalisation and Computer Human Adapted Interaction with a strong focus on Education applications. Her work focuses on mining users' data for building smart, personalised solutions as well as on the creation of novel and adaptive interfaces for supporting users' tasks. She regularly serves on the program committees of international conferences and editorial boards in the fields of Artificial Intelligence in Education and Educational Data Mining and she is the editor of the Journal on Educational Data Mining.

Judy Kay is Professor of Computer Science at the University of Sydney and a principal on the Computer Human Adaptive Interaction, CHAI lab. She received her PhD in Computer Science from University of Sydney in 1998. This paper builds from her body of personalisation research, which aims to exploit the huge amounts of data available about people, from conventional and emerging systems, to create user models and personalised systems. She has been Programme Chair of leading conferences and had Editorial Board roles for the key journals in Personalisation, Pervasive and Ubiquitous Computing and Artificial Intelligence in Education.