People Recommendation Based on Aggregated Bidirectional Intentions in Social Network Site

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

School of Computer Science and Engineering,
The University of New South Wales, Sydney, NSW, 2052, Australia
{yskim,ashesh,compton,xcai,mike,alfredk,wobcke}@cse.unsw.edu.au

Abstract. In a typical social network site, a sender initiates an interaction by sending a message to a recipient, and the recipient can decide whether or not to send a positive or negative reply. Typically a sender tries to find recipients based on his/her likings, and hopes that they will respond positively. We examined historical data from a large commercial social network site, and discovered that a baseline success rate using such a traditional approach was only 16.7%. In this paper, we propose and evaluate a new recommendation method that considers a sender's interest, along with the interest of recipients in the sender while generating recommendations. The method uses user profiles of senders and recipients, along with past data on historical interactions. The method uses a weighted harmonic mean-based aggregation function to integrate "interest of senders" and "interest of recipients in the sender". We evaluated the method on datasets from the social network site, and the results are very promising (improvement of up to 36% in success rate).

Keywords: Recommender systems, Social networks.

1 Introduction

A traditional recommender system considers a user's preferences along with characteristics of objects being recommended to suggest objects that a user might be interested in. Here, objects could be books, music albums, consumer goods, etc. It is not always easy to develop a good recommender system because such a system should take into account the hidden preferences of a user and often the special characteristics of objects. Such systems are based on one-way interaction model that assumes that objects that are being recommended are not active, and therefore the "intentions" of an object are not considered while making recommendations. However, when recommending people to a given user in a social network site, we also need to consider whether a person recommended is likely to "like" a given user or not. In other words, here objects being recommended are not "passive", and importantly we need to consider the "intentions" of a person being recommended to a given user, along with the "intentions" of a given user. We call this kind of interaction a two-way interaction model. A recommender system for a two-way interaction model needs to consider bidirectional intentions while generating recommendations.

B.-H. Kang and D. Richards (Eds.): PKAW 2010, LNAI 6232, pp. 247–260, 2010.

In this research, we propose a new recommendation method for a two-way interaction model. We classify users into two types – senders and recipients – according to their behaviours. For each sender, the method creates a recommendation rule using profiles of senders and recipients, along with interaction history of all users. The method combines the "*interests of senders*" and the "*interest of recipients in the senders*" using a criteria *aggregation function*. The importance of these two interests is represented by weights. These weights could be heuristically or experimentally decided to create effective recommendations. Experimental results show that the acquired rules are significantly different for different weights for "*interests of senders*" and "*interest of recipients in the senders*". We classified users into four groups based on the number of interactions they have received, and analyse their performance.

This paper is organized as follows: Section 2 summarizes the literature review on recommender systems and multi-criteria aggregation research. Section 3 explains our rule learning method and Section 4 explains the evaluation method. Experimental results are summarized in Section 5. Conclusions and further work are discussed in Section 6.

2 Literature Review

2.1 Recommender Systems

Recommender systems usually propose candidate items/objects to a given user by using similar users' preference patterns. For example, if user 1 and user 2 are similar and user 1 purchases item A, the system suggests item A to user 2. Here, defining similarity of users is a critical problem that a recommender system must address. Often user profiles and behaviour are used to define similarity. **Profile-based methods** use user profiles to calculate similarity between users. Various machine learning techniques, such as decision trees, rule induction, nearest neighbour, and Naïve Bayes classification, are usually employed for this purpose [1]. Profile-based recommenders may not work well when user profiles are not sufficient for learning user similarity[2]. **Behaviour-based methods** use users' behaviour to calculate similarity. For example, the Amazon recommender system [3] uses view or purchase history to identify similar users. Many other item-to-item collaborative filtering recommender systems [4-6] and social network-based recommenders [7-10], use users' behaviour to identify user similarity. We employ behaviour-based methods as well as profile-based methods. The social network site we examined provides us with a large dataset outlining user profiles for a very large number of users, and past interactions between users.

2.2 Bidirectional Criteria in Recommender Systems

As discussed earlier, we need to consider bidirectional intentions for our recommendations. It is important to consider how to combine two different "*intentions*", before making recommendations. Though multi-criteria decision making has been extensively researched [11], it has not received attention in the recommendation research community. Several researchers discussed multi-criteria in relation to rating problems of collaborative filtering based recommenders [12-14]. Even though our research is not based on the collaborative filtering method, it is necessary to reflect two different intentions, namely the intentions of senders and the recipients and thus it is necessary to consider their aggregation. Aggregation functions could be obtained by domain expertise, statistical techniques, and/or machine learning techniques [12]. We used a weighted harmonic mean based aggregation function, because "*interest of senders*" and "*interest of recipients in the sender*" are ratios and typically a weighted harmonic mean is appropriate for situations where a weighted average of rates is desired.

3 Recommendation Rule Acquisition Method

3.1 Definitions

A **user** is represented by M attribute values. A **sender** is a user who initiates an interaction and a **recipient** is a user who receives an interaction from a sender.

A **subgroup** is a group of users where at least m attribute have the same values, here $1 \leq m \leq M$. A **basic subgroup** is a subgroup where m equals 1. For example, the set of users belonging to the same ethnic group is a subgroup.

An **interaction** is an action where a sender sends a message to a recipient. A **sending interaction** is an interaction where a sender has not received an interaction from the recipient. In other words, a sender is initiating a new conversation. A sending interaction is represented by an arrow (→). A **responding interaction** is an interaction where a sender has already received an interaction from the recipient. In other words, this represents a reply from a recipient of an interaction. A responding interaction is represented by an arrow (←). The number of sending interactions from a subgroup of senders (S_i) to all the recipients (R) is denoted as $\mathrm{ns}(S_i \rightarrow R)$, and the number of sending interactions from a subgroup of senders (S_i) to a subgroup of recipients (R_i) is denoted as $\mathrm{ns}(S_i \rightarrow R_j)$. Whereas a sending message usually shows only a positive intention of a sender, a responding message exhibits both positive and negative intentions of a recipient. The number of positive responding interactions from a subgroup of recipients (R_i) to a sender subgroup (S_i) is denoted as $\mathrm{ns}(\mathrm{S_i} \leftarrow \mathrm{R_j}(+))$. Similarly, the number of negative responding interactions from a subgroup of recipients (R_i) to a sender subgroup (S_i) is denoted as $\mathrm{ns}(\mathrm{S_i} \leftarrow \mathrm{R_j}(-))$.

Interest of Senders: For a sender subgroup ($\mathrm{S_i}$), its interest in a recipient subgroup ($\mathrm{R_i}$) is defined as follows:

$$\mathrm{I}(\mathrm{S_i}, \mathrm{R_j}) = \mathrm{ns}(\mathrm{S_i} \rightarrow \mathrm{R_j})/\mathrm{ns}(\mathrm{S_i} \rightarrow \mathrm{R}), \tag{1}$$

where $\mathrm{ns}(\mathrm{S_i} \rightarrow \mathrm{R_j})$ represents the number of interactions sent from the sender subgroup ($\mathrm{S_i}$) to the recipient subgroup ($\mathrm{R_j}$) and $\mathrm{ns}(\mathrm{S_i} \rightarrow \mathrm{R})$ represents the number of interactions sent from the sender subgroup ($\mathrm{S_i}$) to all recipients (R). This measure represents how much the sender subgroup ($\mathrm{S_i}$) is interested in the recipient subgroup ($\mathrm{R_j}$), compared to the rest of the recipients in R.

Interest of Recipients in Senders: For a recipient subgroup (R_j), its interest in a sender subgroup (S_i) is defined as follows:

$$I(R_j, S_i) = ns(S_i \leftarrow R_j(+))/ns(S_i \rightarrow R_j), \tag{2}$$

where $ns(S_i \rightarrow R_j)$ represents the number of interactions sent from the sender subgroup (S_i) to the recipient subgroup (R_j) and $ns(S_i \leftarrow R_j(+))$ represents the number of positive responses sent from the recipient subgroup (R_j) to the sender subgroup (S_i). This measure represents how much the recipient subgroup (R_j) is interested in the sender subgroup (S_i).

3.2 Interaction Look-Up Table

As we modelled our recommendation method using profiles and behaviours of users, each user's profile and the log of interactions between users were collected from a specified training period. Based on these data, an interaction look-up table for each attribute was created for rule learning. Fig. 1 illustrates such an interaction look-up table for a single attribute.

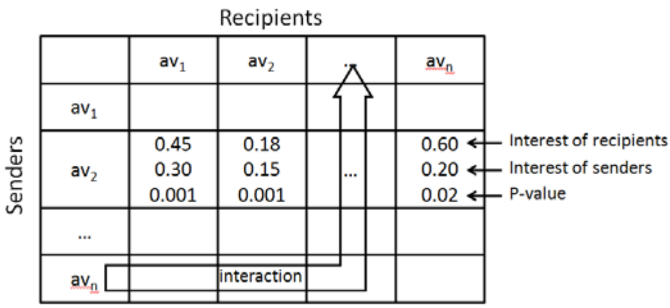


Fig. 1. Interaction Look-up Table between Attribute Values

Each row represents senders with a particular attribute value, and each column represents recipients with a particular attribute value. For example, for an interaction look-up table for the attribute "Ethnicity", different rows would represent senders of different ethnicities (Greek, Vietnamese, English, etc.), and similarly different columns would represent recipients of different ethnicities. In this interaction look-up table, each cell contains *interaction measures* between the senders in the corresponding row and the recipients in the corresponding column. There is a major simplifying assumption in this, that attributes are independent and that the interaction between senders and recipients can be considered separately for each attribute.

3.3 Best Matching Pair Discovery

For each attribute of a given sender, the method attempts to find a best matching value of the same attribute for recipients, such that the sender is likely to be interested in the recipients, and the recipients are more likely to reply positively. For example, "Australian" senders send 80% of their interactions to "English" recipients, and 20% of their interactions to "German" recipients. This clearly indicates that "Australian" senders are more interested in "English" recipients than "German" recipients. Now, suppose 40% of the "English" recipients reply positively, and 70% of the "German" recipients reply positively. If we want to recommend recipients based on the "interest of senders" (see Section 3.1), we should recommend "English" recipients. However, if we want to recommend recipients based on the "interest of recipients in senders", we should recommend "German" recipients. This example illustrates that each criterion only partially captures the interest of the sender and recipients. We need to combine these two criteria in order to generate recommendations such that both a sender and the corresponding recipients are interested in each other.

In this research, we used the following *interest aggregation* function to integrate a sender's interest in recipients ("*interest of senders*", see Section 3.1) and recipients' interest in senders ("*interest of recipients in the sender*", see Section 3.1).

Interest Aggregation Function: A weighted harmonic mean, $H(S_i, R_j)$, was employed as an *interest aggregation* function, which is defined as follows:

$$H(S_i, R_j) = \frac{\omega_s + \omega_r}{\frac{\omega_s}{I(S_i, R_j)} + \frac{\omega_r}{I(R_j, S_i)}} \tag{3}$$

where $I(S_i, R_j)$ and $I(R_j, S_i)$ represent the interest of senders and the interest of recipients in senders respectively (see Section 3.1), ω_s represents a weight for "*interest of senders*" and ω_r represents a weight for "*interest of recipients in the sender*". The sum of these two weights ($\omega_s + \omega_r$) is 1.

For a given sender and a value of ω_s, the method calculates weighted harmonic mean for each value of the same attribute for recipients using Eq (3). The method selects an attribute value for recipients with the highest weighted harmonic mean as the best matching pair for that attribute. In our experiments, various combinations of the weights were evaluated.

3.4 Recommendation Rule Acquisition

For a given sender, the method finds best matching pairs for every attribute. From these best matching pairs, all the recipient attribute values can be collected to formulate a rule that could be used to generate possible recommendations. For example, the following rule (only few conditions are displayed here) could be used to generate recommendations for a given sender,

```
Gender = Female
AND Job = Accounting
AND Location = Adelaide
AND Age = 40 ~ 44
. . . .
. . . .
```

However, in practice such a rule with all the recipient attribute values may prove too specific and may not generate any recommendations. If such a rule could not generate the required number of recommendations, the method relaxes the rule by removing an attribute from a rule that has the lowest value for "*interest of recipients in senders*".

The process is repeated and the rule is relaxed until we can generate sufficient recommendations to satisfy the constraints discussed in Section 3.5.

In the process of selecting an attribute for relaxing a current rule, we did not use criteria such as "*interest of senders*" (see Section 3.1) and "*interest aggregation value*" (see Section 3.2) because they are influenced by the number of possible values of an attribute, and therefore we could not compare them across different attributes.

3.5 Constraints for Candidate Generation

We are only interested in candidates who are active users. We define an active user in three ways:

- The user joined the social networking website recently,
- Or the user sent initiating interactions recently,
- Or the user received and viewed initiating interactions from others recently. (The log identifies if a recipient looks at a contact message).

We define 'recently' as the last month. Preliminary data analysis on temporal activity showed that user activity in the past month provides a good indication of how likely they are to respond.

We are also interested in discovering subgroups of senders and recipients such that the interaction behaviour between them is statistically significant. We could use the current rule to generate a subgroup of recipients, and the corresponding subgroup of senders (based on the corresponding best matching pairs). We calculate the probabilistic significance of the interaction behaviour between these two subgroups using the following binomial formula:

$$\mathbf{P(r)_{binomial} = nCr \times p^{r} \times q^{n-r}} \tag{4}$$

where n is the number of sending interactions between the two groups, r is the number of positive replies between the two groups, p is the base success rate between all senders and recipients (no of all positive replies/no of all sending interactions), q is 1-p. We consider an interaction behaviour is significant if the p-value is < 0.05.

4 Experimental Design

4.1 Data Sets

We used data sets obtained from a large commercial social network site. User profile data contained 32 attributes for each user, such as age, location, ethnic background, physical appearance (body type, hair colour, etc.), occupation industry and level, children and marital status, and others. All numeric attribute values such as age, number of photos, number of children, etc were transformed to nominal values. User interaction log data contains the interaction history between senders and its corresponding recipients. Each log entry identifies a sender, the corresponding recipient and the reply message. Reply messages are classified into positive and negative, so that each interaction is also classified as a positive or negative interaction. A failure to reply was taken as a negative interaction. This interaction log data is used to calculate interest measurers between a sender subgroup and its corresponding recipient subgroup (see Section 3.1). Two sets of training data, **Train I** and **Train II**, and one set of test data, **Test**, were collected for the experiment. The data sets used for this research are summarised in Table 1. **Train I** was used for our rule learning method. It contains interaction history data for three months and was used to generate the interaction look-up tables and also to calculate "interest of senders" (see Section 3.1), "interest of recipients in the senders" (see Section 3.1), and p-value for significance test (see Section 3.5) in the rule learning process. **Train II** was collected for the Collaborative Filtering (CF)-based method from March, 2009 (one month). Our preliminary data analysis using the CF method over different time periods showed that a training period of one month was appropriate. **Test** data were collected from the first week of April for evaluation. This was immediately following the CF training period to give the CF method the best chance of performing.

Table 1. Training and Test Data Set

Data Set	Total interaction	Positive Interaction	Negative Interaction	Success Rate
Train I	*3,888,034*	*689,419*	*3,198,615*	*17.7%*
Train II	*1,357,432*	*236,521*	*1,120,911*	*17.4%*
Test	*284,702*	*47,468*	*237,234*	*16.7%*

We examined whether or not the number of interactions received influences recommendation performance and recommendation rule acquisition. All senders in the test period (30,387) were classified into four types of senders based on the number of interactions received during March, 2009. Note that zero-received sender in the training period may receive interactions in the test period.

Table 2. Sender Types Based on Interaction Received

Sender Type	Interaction Received (n)	Users (%)
zero-received	$n = 0$	7,560 (25%)
few-received	$1 \leq n \leq 3$	8,507(28%)
Average	$4 \leq n \leq 20$	11,223(37%)
Popular	$20 < n$	3,097(10%)

Recommendations were generated using different weights for "*interest of senders*" (ω_s) and "*interest of recipients in the sender*" (ω_r). We used 0, 0.25, 0.50, and 0.75 for ω_r and the corresponding ω_s is 1.0, 0.75, 0.5, and 0.25, as $\omega_s + \omega_r$ is 1 (see Section 3.3). In the following discussion, unless otherwise indicated, "*weight*" means "*interest of recipients in the sender*" (ω_r). Weight 0 was employed to evaluate an extreme situation, where only "*interest of senders*" is considered for rule learning. Weight 0.5 was employed to evaluate when the two interests are regarded to have same importance. Weight 0.25 and weight 0.75 (ω_r equal 0.25) were selected to see the results if one of the two interests has more importance than another. Weight 1.00, which is another extreme case that considers only "*interest of recipients in the sender*", was not employed, because senders in the testing period rarely sent interactions to the recipients recommended by our method when weight is 1.00. This is understandable because weight 1.00 does not consider "*interest of senders*".

4.2 Collaborative Filtering

Interactions are likely to depend significantly on an individual's appearance in photos and other personal preferences which may be included in free text, but are not captured in the attribute data we used. In this case, Collaborative Filtering (CF) could be used to generate recommendations. We implemented a CF method based on [1] and compared it with our approaches. In a typical CF model, two items are similar if they have been purchased together by a large number of customers, and the unpurchased item is suggested to a user if he/she purchases the other similar item. Based on these criteria, two users are considered to be similar senders to the extent that they have sent contacts to the same recipients. For example, if sender s_1 sends message to recipient r_1, r_2, and r_3 and sender s_2 is to recipient r_1 and r_2, they are regarded as similar senders, because they both sent interactions to recipient r_1 and r_2. If a new user *u* sent contacts to r_1, then u is also similar to s_1 and s_2, so r_1, r_2 and r_3 can be recommended to u. Note that the rank of r_1, r_2 and r_3 may be differ because r_1 and r_2 are recommended by both s_1 and s_2, whereas s_3 is only recommended by s_1. For the evaluation, the test set was checked to see whether the interactions between a user and the candidates suggested by our recommendation methods and the CF method had actually occurred. As this was a retrospective study, we could not assess what would happen if users had followed our recommendations. All we could check was whether they had a higher success rate if they had happened to contact a person we would have recommended. Very popular users, who received more than 50 contacts in March, 2009, were not considered for evaluation.

4.3 Evaluation Metrics

We used the following metrics and variables to assess our method:

- Θ: a given recommendation method,
- *M*: the senders who were active in March, 2009 and who sent interactions in the test period,
- *N*: the senders who were members of *M* and who would have received suggestions by Θ,
- *O*: all interactions suggested by Θ (representing a sender and a predicted recipient) for all senders in N, and
- *Q*: all interactions in the testing period by all senders in *M*, and
- *K*: the intersection between *O* and *Q*

For the performance evaluation of each method, the following metrics were used:

Coverage: The proportion of *N* from *M* in the test period, i.e.

$$Cov = \mathrm{n}(N)/\mathrm{n}(M) \tag{5}$$

where n(*M*) is the number of *M* and n(*N*) is the number of *N*.

Success Rate: The proportion of those predicted successful contacts of K, i.e.

$$Succ = ns(K,+)/ns(K) \tag{6}$$

where $ns(K)$ is the number of interactions of K and $ns(K,+)$ is the number of positive interactions of K.

For the rule acquisition evaluation, the following metrics were used:

Rule Usage: The number of users per rule, i.e.

$$Usage = \mathrm{n}(N)/\mathrm{n}(R) \tag{7}$$

where $\mathrm{n}(N)$ is the number of users covered by a given method and $\mathrm{n}(R)$ is the number of rules from that method.

Condition Complexity: The number of condition elements per rule, i.e.

$$Complexity = \mathrm{n}(C)/\mathrm{n}(R) \tag{8}$$

where $\mathrm{n}(C)$ is the number of condition elements of all rules from given method and $\mathrm{n}(R)$ is the number of rules for that method.

5 Results

5.1 Coverage

On average the CF method can suggest recommendations for 74.0% of all senders in the test period (see Table 3). Whereas the CF method could suggest many recommendations for the senders who received many interactions in the recent month (March, 2009), it could suggest fewer recommendations for the senders who did not receive many interactions in the recent month. It suggests recommendations for 77.3% of the *few-received* senders, 86.7% of *average* senders, and 89.2% of *popular* senders in the testing period, but it only suggests recommendations for 45.4% of *zero-received* senders. However, our method can generate recommendations for all the senders in the testing period.

Table 3. Recommended Candidates Ratios

Sender Type	CF (Coverage)	Rule (Coverage)
Zero-received	*3,431 (45.4%)*	7,560 (100.0%)
Few-received	6,572 (77.3%)	8,507 (100.0%)
Average	9,725 (86.7%)	11,223 (100.0%)
Popular	2,767 (89.2%)	3,097 (100.0%)
All users	22,495 (74.0%)	30,387 (100.0%)

5.2 Success Rate

Success rate results are summarized in Table 4. For the test period interactions, the overall baseline success rate is 16.7%. *Popular* senders (24.8%) show the highest baseline success rate, followed by *average* senders (18.2%), *zero-received* senders

(14.1%), and *few-received* senders (14.0%). In some way, as the number of received interactions represents the popularity of the senders, this result is consistent with common sense that popular users receive more positive replies than less popular users.

Table 4. Success Rate

Sender Types	Test	CF	Rule with different weight for ω_r 0.00	0.25	0.50	0.75
Zero-received	*14.1%*	*11.0%*	*18.1%*	*18.5%*	*18.9%*	*20.0%*
Few-received	*14.0%*	*13.8%*	*20.2%*	*21.3%*	*20.9%*	*20.6%*
Average	*18.2%*	*19.3%*	*23.6%*	*24.0%*	*24.8%*	*25.3%*
Popular	*24.8%*	*25.8%*	*30.8%*	*30.4%*	*30.5%*	*30.3%*
All users	*16.7%*	*17.3%*	*21.5%*	*22.0%*	*22.2%*	*22.6%*

The success rate of the CF method (17.3%) is slightly higher than that of the testing baseline success rate. Similarly it is slightly higher for popular and average senders, and more significantly lower for zero-received. Our method gives higher success rates for all types of users (21.5% ~ 22.6%) (see Table 4), compare to both the baseline rates and the CF method. This is caused by the fact that whereas the CF method only reflects "interests of senders", our method reflects both "interests of senders" and "interests of recipient in the sender". The success rates increase as the weights increase from 0.00 to 0.75, but it is not significant. Success rate improvement for each sender types is summarized in Table 5, where success rate improvement values obtained by dividing each success rate by its corresponding baseline success rate of the testing period. The results show that our method is more effective for the "*few-received*" senders, followed by "*zero-received*", "*average*", and "*popular*" senders. This result is interesting because less popular users generally need more help from recommender systems than popular users.

Table 5. Success Rate Improvement

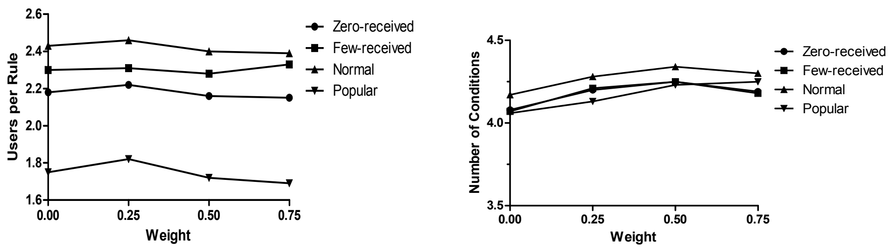
Sender Types	Test	CF	Rule with different weight for ω_r 0.00	0.25	0.50	0.75
Zero-received	*1.00*	*0.78*	*1.28*	*1.31*	*1.34*	*1.42*
Few-received	*1.00*	*0.98*	*1.44*	*1.52*	*1.49*	*1.47*
Average	*1.00*	*1.06*	*1.30*	*1.32*	*1.36*	*1.39*
Popular	*1.00*	*1.04*	*1.24*	*1.22*	*1.23*	*1.22*
All users	*1.00*	*1.04*	*1.29*	*1.32*	*1.33*	*1.36*

5.3 Rule Acquisition

Our method generates a rule for each user to produce the most appropriate recommendations, so the number of rules should be the same as the number of senders in the testing period. However, as the same rules can be generated for different senders, far fewer rules are created as summarized in Table 6. The average number of users per rule is illustrated in Fig. 2 (a), showing less "*popular*" senders and "*zero-received*" senders per rule than "*few-received*" and "*normal*" senders. This means more rules are created for "*popular*" senders and "*zero-received*" senders compared to "*few-received*" and "*normal*" senders. This may be caused by the fact that the rules for "*popular*" senders and "*zero-received*" senders may be more complex than others. However, the data does not show that the rules for "*popular*" senders and "*zero-received*" senders use more conditions than rules for other types of users (see Fig. 2 (b)).

Table 6. Rule Acquisition Results

Sender Types	Rule with different weight for ω_r			
	0.00	0.25	0.50	0.75
Zero-received	3,470	3,408	3,503	3,509
Few-received	3,705	3,684	3,733	3,656
Average	4,617	4,557	4,686	4,697
Popular	1,767	1,702	1,800	1,830
All users	8,910	8,538	8,766	8,685



(a)Average Users per Rule (b)Conditions per Rule

Fig. 2. Rule Acquisition

Table 7 shows some example users where different weights produce different rules. For example, for the user id 1074, three different rules (rule ids 24, 14574, 23897) are created to recommend possible recipients. Taking user 1074 as an example, the first column in Table 8 lists some of the attribute values of the sender, and the second column lists three different rules (based on different weights) for generating possible recommendations. When the weight is zero, female recipients who live in "**Adelaide**" and whose job is in "**Healthcare / Medical**" are proposed as recommendations. As shown in Table 9, the senders who live in "**Adelaide**" sent 84% of their interactions to recipients in "**Adelaide**", and 30% of these interactions result in positive replies. Similarly, senders who "**have children living at home sometimes**" sent 34% of their interactions to recipients who "**have children living at home**" and 32% of these interactions result in positive replies. When the weight is zero, the method only considers sender's interest and ignores recipients' interest in senders. Therefore, the selection of attributes is biased towards higher "*interest of senders*" values. For example, **location=Adelaide** is selected due to its higher "Interest of Senders" value, when ω_r=0. However, when the weight is 0.75, more emphasis is given to the values of "*interest of recipients in the sender*" while selecting attributes. For example, "**`location=Adelaide`**" is not selected due to its lower "*interest of recipients in the sender*" value compare to other three attributes listed in the rule for ω_r=0.75.

Table 7. Examples of the Recommended Rule Changes

User ID	Applied Rule ID with different weight for ω_r			
	0.00	0.25	0.50	0.75
98	3	3	3	23892
462	8	14562	14562	23894
735	13	14	14	14
1074	**24**	**14574**	**14574**	**23897**
1166	26	14578	19670	23899
1364	28	14581	19673	23900

Table 8. Example Rules of User 1074

Senders		Rules for recommended recipients
	Rule ID	Condition

`Gender = Male`
`Age = 40~44`
`Job = Property/Real Estate`
`Location = Adelaide`
`Have Children = Yes, have children living at home sometimes`
	24
	(ω_r=0)
		`Gender = Female`
		`AND Location = Adelaide`
		`AND Job = Healthcare / Medical`

	14574
	(ω_r=0.25 or 0.5)
		`Gender = Female`
		`AND Have Children=Yes, have children living at home`
		`AND Job = Healthcare / Medical`
		`AND Location = Adelaide`

	23897
	(ω_r=0.75)
		`Gender = Female`
		`AND Have Children=Yes, have children living at home`
		`AND Job = Healthcare / Medical`
		`AND Age = 40 ~ 44`

Table 9. Interest of Senders and Interests of Recipients in the Sender

Attribute	Attribute Value		Interest of Senders	Interest of Recipients in the Sender
	Male	Female		
Location	Adelaide	Adelaide	84%	30%
Have Children	Yes, have children living at home sometimes	Yes, have children living at home	34%	32%
Age	40~44	40~44	24%	31%
Job	Property / Real Estate	Healthcare/ Medical	12%	31%

6 Conclusion and Further Work

Recommendations for users in social network sites should reflect both "*interests of senders*" and "*interest of recipients in the sender*". We propose a new method that combines these two interests using a weighted harmonic mean. Various combinations of weights for these two interests were examined in the experiments. Experimental results show our method obtains significantly higher success rate, up to 36% higher than the base line success rate of 16.7%. We also observed that the success rates differs significantly for different types of users (see Table 5).

As this experiment is a retrospective study, we could only measure success rates on the historical interactions that were initiated by senders. This also means that such interactions incorporate senders' interest, because recipients were selected by senders. This also explains why overall success rates for different weight values are not very different. However, in future we plan to study how a sender's activity is influenced by our recommendations. In particular, how users respond to recommendations generated by different weight values.

In future, we also plan to investigate alternative rule generation methods. For example, relaxing a rule by relaxing possible values an attribute could be assigned to. For example, location= "**`Sydney City`**" could be relaxed by location= "**`Sydney City`**" `OR` "**`Sydney North`**". This would allow us to generalise a rule using small incremental steps. We could also use domain knowledge in selecting or rejecting certain attributes in the process of rule creation. For example, we could consider attributes like location, age, etc to be very important and therefore they should be included in a final rule. We could also consider using different weights for calculating weighted harmonic-mean values for different types of users based on their past activities.

Currently we do not rank recommendations generated by a rule. In future, we plan to rank recipients, for a given rule, based on their likelihood of replying positively to a given sender. This could be very useful if only a small number of recommendations are made. We would also like to explore alternatives to our simplifying assumption that each attribute can be considered separately in calculating preferences between senders and recipients.

Acknowledgments. This research has been supported by the Smart Services Cooperative Research Centre.

References

1. Pazzani, M.J., Billsus, D.: Content-Based Recommendation Systems. In: Brusilovsky, P., Kobsa, A., Nejdl, W. (eds.) Adaptive Web 2007. LNCS, vol. 4321, pp. 325–341. Springer, Heidelberg (2007)
2. Balabanović, M., Shoham, Y.: Fab: Content-Based, Collaborative Recommendation. Communications of the ACM 40(3), 66–72 (1997)
3. Linden, G., Smith, B., York, J.: Amazon.Com Recommendations: Item-to-Item Collaborative Filtering. IEEE Internet Computing 7(1), 76–80 (2003)
4. Deshpande, M., Karypis, G.: Item-Based Top-N Recommendation Algorithms. ACM Transactions on Information Systems (TOIS) 22(1), 143–177 (2004)
5. Karypis, G.: Evaluation of Item-Based Top-N Recommendation Algorithms. In: Tenth International Conference on Information and Knowledge Management, pp. 247–254 (2001)
6. Sarwar, B., Karypis, G., Konstan, J., Reidl, J.: Item-Based Collaborative Filtering Recommendation Algorithms. In: 10th International Conference on World Wide Web, pp. 285–295 (2001)

7. Huang, Y., Contractor, N., Yao, Y.: Ci-Know: Recommendation Based on Social Networks. In: 9th Annual International Digital Government Research Conference, pp. 375–376 (2008)
8. Mobasher, B., Burke, R., Bhaumik, R.: Attacks and Remedies in Collaborative Recommendation. IEEE Intelligent Systems 22(3), 56–63 (2007)
9. Palau, J., Montaner, M., de la Rosa, J.L.: Collaboration Analysis in Recommender Systems Using Social Networks. In: The Third International Joint Conference on Autonomous Agents and Multi Agent Systems, pp. 137–151 (2004)
10. Perugini, S., Gonçalves, M.A., Fox, E.A.: Recommender Systems Research: A Connection-Centric Survey. Journal of Intelligent Information Systems 23(2), 107–143 (2004)
11. Figueira, J.R., Greco, S., Ehrgott, M.: Multiple Criteria Decision Analysis: State of the Art Surveys. International Series in Operations Research & Management Science, vol. 78 (2005)
12. Adomavicius, G., Kwon, Y.: New Recommendation Techniques for Multicriteria Rating Systems. IEEE Intelligent Systems 22(3), 48–55 (2007)
13. Lee, H.-H., Teng, W.-G.: Incorporating Multi-Criteria Ratings in Recommendation Systems. In: IEEE International Conference on Information Reuse and Integration (IRI 2007), pp. 273–278 (2007)
14. Rattanajitbanjong, N., Maneeroj, S.: Multi Criteria Pseudo Rating and Multidimensional User Profile for Movie Recommender System. In: 2nd IEEE International Conference on Computer Science and Information Technology, ICCSIT 2009, pp. 596–601 (2009)