# Reciprocal Recommendation for Job Matching with Bidirectional Feedback

Tsunenori Mine[†,‡] Tomoyuki Kakuta[†] Akira Ono[†]
*Department of Advanced Information Technology, {Graduate School[†], Faculty[‡]} of ISEE, Kyushu University, Fukuoka 819-0395, Japan, Email: mine@ait.kyushu-u.ac.jp ‡*

***Abstract*—This paper proposes a novel reciprocal recommendation method for job matching with bi-directional feedback. The proposed method uses, as mutual feedback, bilateral messages between job seekers and recruiters, such as applying to a job, scout of a seeker, and reply to the offer, on the seekers-recruiters' user network. During job matching process, user agents, as delegate of their owners, send and receive those messages with each other. From those feedback messages, each user agent computes the popularity degree of its owner user: seeker or recruiter, and evaluation degree of each other from the popularity degree; considering both the popularity and evaluation degrees, and the similarity between a condition provided by its user and a profile of each candidate user, the agent dynamically updates a ranking list for recommendation of its owner user after every matching action. Preliminary experiments illustrate the validity of the proposed method.**

***Keywords*-Job Matching, Reciprocal Recommendation, MultiAgents**

## I. Introduction

Job matching is one of socially crucial problems in all over the world. In job matching, job seekers and recruiters have requirements for the others, apply to and scout them, and proceed further matching actions according to the preference order of their candidates. Matching actions with self-knowledge and care for requirements of others make the matching success rate increase and the whole cost of matching decrease. Therefore, the key to improve success rate of matching and to decrease matching costs are to establish a matching method considering mutual desires and requirements. A matching considering reciprocal demand is called two-sided matching. A lot of theoretical studies about stable matching problems[1] have been conducted for more than 50 years. Although the problem setting assumes every participant of matching submits his/her preference order list, it is regarded as one of important issues to create the preference list in real situations. In addition, recommending matching candidates to a user is also one of important research issues. Considering job matching, the stable matching algorithm produces optimal matching for either a job seeker or a recruiter. It does not always reflect reciprocal demand of users.

These days, reciprocal recommendation methods have been proposed (e.g. [2]). The methods aim to give recommendation that improves success rate of matching between men and women by considering their reciprocal interests. The recommendation methods select attributes extracted from online dating site data so that they can consider reciprocal interests. Experimental results show the effectiveness of the reciprocal recommendation methods(e.g. [2], [3], [4]). However, these studies assume that for two users $A$ and $B$, recommending $B$ to $A$ becomes successful if $A$ takes the recommendation, gives an offer to $B$, and $B$ returns positive response to $A$. The positive response does not mean acceptance to go out with or of marriage, but just acceptance of meeting or of exchanging contact information. When considering job matching, there are many people who get the chance to go to the first interview stage, but do not proceed to the final interview stage, or do not get the official offer. Therefore, reciprocal recommendation for job matching should consider the whole process of job matching.

For job matching, Malinowski et al. makes the point the necessity of recommendation process considering reciprocal demand and defines the job matching problem as Pareto-optimization problem[5]. Hosono et al. proposed a kind of two-sided matching method that uses lists of candidates of job seekers and recruiters, and makes the job seekers apply to the recruiters, who are on the list of the job seekers' candidates and whose candidate lists include the job seekers, and vice versa[1]. Unfortunately, they only use the profiles of job seekers and recruiters and do not consider any information found during matching process. Ladasearch [2] takes a dialogical approach called laddering, which is often used for marketing survey, to extract such attributes or other information of job seekers that help to fill the gap between conditions of recruiters and profiles of job seekers. Although the attributes or other information extracted by Ladasearch may be useful and important for job matching, it is not clear if the information is used for matching of the other users.

Considering the previous studies, in this paper, we propose a novel reciprocal recommendation method for job matching. The method uses users' actions and response information as positive or negative information. The positive and negative information can be regarded as feedback information for other user. By using the bi-directional feedback information exchanged, the method updates the similarity score between users. This method has an analogy of link analysis algorithms of Web pages(e.g. [6]) or citation analysis algorithms. The difference point between the proposed method and the

[1]Japanese published unexamined application No. 2009-301422
[2]http://www.oki.com/jp/rd/language/laddering/download/ ladasearch.pdf

978-0-7695-5071-8/13 $26.00 © 2013 IEEE
DOI 10.1109/IIAI-AAI.2013.91

other previous work is that the proposed method considers the job matching process where a user can expect either positive or negative reply of the other user applied to by the user; on the other hand, an Web page or a paper does not expect any reply from its linked Web page or cited paper.

In what follows, Section II describes the proposed method; Section III discusses an evaluation method by simulation and shows some of highlighted results, and lastly we conclude in Section IV.

## II. Job Matching with Bi-directional Feedback

### A. Preliminary

This section describes terminologies used by the proposed method.

- User: We call a person who has the role of either job seeker or recruiter, a user. If two users have the same role, we call them homogeneous users, and otherwise, heterogeneous users.
- User Condition and Profile: for user $X$ and $Y$, the profiles of user $X$ and $Y$ are described as $P_X$ and $P_Y$, respectively. A condition from $X$ to $Y$ is $C_{XY}$ or just $C_X$ if $Y$ is clearly determined or has nothing to say. We assume that $A$ is a set of $(a, v_a)$ which is a pair of user attribute $a$ and its value $v_a$. A user profile $P$ and condition $C$ are defined as follows: $P_i \subseteq A(i \in \{X, Y\})$, $C_i \subseteq A(i \in \{X, Y\})$
- SCP (Similarity between Condition and Profile): For user $X$ who actively sends a query and finds its matching candidate $Y$, $Sim_{CP}(C_X, P_Y)$ is defined as a similarity between user $X$'s condition $C_X$ and user $Y$'s profile. $Sim_{CP}(C_X, P_Y)$ is defined as follows: $Sim_{CP}(C_X, P_Y) =$
$$\sum_{a \in A} \delta(a) w_{X,a} Sim_{Atr}(a_{C_X}, a_{P_Y}) \quad (1)$$
Function $\delta$ judges if $a$ is a common attribute of both $C_X$ and $P_Y$; $\delta(a) = 1$ if $a \in f_X$ and $a \in f_Y$ $(f \in \{P, C\})$, 0 otherwise. $w_{X,a}$ is a weight of attribute $a$ used by user $X$, and is determined by the ratio between the total number of $a$ and the total number of all attributes that have been appeared in $C_X$ and $P_Y$ so far.
$$\begin{aligned} w_{X,a} &= \left(1 + \frac{c_s(X,a)}{\sum_{a' \in A} c_s(X,a')}\right) \\ &\times \left(1 + \left(\frac{c_a(X,a)}{\sum_{a' \in A} c_a(X,a')} - 0.5\right)\right) \\ &\times \left(1 + \left(\frac{c_q(X,a)}{\sum_{a' \in A} c_q(X,a')} - 0.5\right)\right) \end{aligned}$$
$c_s(X, a)$ is the total number of times that user $X$ used attribute $a$ in search; $c_a(X, a)$ and $c_q(X, a)$ are the maximum number of times that $X$ used attribute $a$ when applying to or asking a question to a matching candidate, respectively.
- SP (Similarity between Profiles) and SC (Similarity between Conditions): for homogeneous user $X_i$ and $X_j$ $(i \neq j)$, the profile and condition similarity between $X_i$ and $X_j$ are respectively described $Sim_P(P_{X_i}, P_{X_j})$ and $Sim_C(C_{X_i}, C_{X_j})$, and defined as follows: $Sim_f(f_{X_i}, f_{X_j}) =$
$$\sum_{a \in A} \delta(a) w_{X_i,a} Sim_{Atr}(a_{f_{X_i}}, a_{f_{X_j}}) \quad (2)$$
$Sim_{Atr}(a_{f_{X_i}}, a_{f_{X_j}})$ is a similarity between attribute $a$ in $f_{X_i}$ and $f_{X_j}$ $(f \in \{P, C\})$.
- SHoU (Similarity between Homogeneous Users): $Sim_{ho}(X_i, X_j)$ is a similarity between homogeneous user $x_i$ and $x_j$, and is defined as follows: $Sim_{ho}(X_i, X_j) =$
$$\alpha Sim_C(X_i, X_j) + (1 - \alpha) Sim_P(X_i, X_j) \quad (3)$$
In the experiment described later, we chose $\alpha = 0.5$.
- SHeU (Similarity between Heterogeneous Users): $Sim_{he}(X_i, Y_j)$ is a similarity between heterogeneous user $X_i$ and $Y_j$ and is defined as follows:
$$Sim_{he}(X_i, Y_j) = Sim_{CP}(C_{X_i}, P_{Y_j}) * Eval(Y_j, X_i) \quad (4)$$
Function $Eval(Y_j, X_i)$ returns an evaluation degree value of $Y_j$ to $X_i$, and is precisely described in Section II-D.

### B. Similarity between Attribute Values

User attributes used in this paper are shown in table I. Each attribute, except for **Salary, Work Hours**, and **Employment Pattern**, is represented by a hierarchical tree with depth $N$. Here, $Sim_{Attr}(a_i, a_j)$ is a similarity between a target node $a_i$ and a candidate node $a_j$ in the tree, and is defined as follows: $Sim_{Attr}(a_i, a_j)$ is 1 if $a_j$ is the same as $a_i$, or a descendent node of $a_i$, $1 - \frac{m}{N}$ $(0 \leq m \leq N)$ if $a_j$ is the same node as $a_m$, which is the $m$-th former ancestor of $a_i$, or the descendent of $a_j$, 0 otherwise. In the experiments described later, $N$ takes 3 or 4.

Table I
User Attribute

| User | Attribute |
|---|---|
| Job Seeker | Academic Background, Job History |
| Recruiter | Category of Business, Job Title, Place of Business, Salary, Work Hours, Employment Pattern |

**Category of Business**: According to the classification of SB-MIC [3], the Category of Business consists of 4 layers: traditional industry categorization, and large, middle, and small classification.

[3]Statistics Bureau, Director-General for Policy Planning and Statistical Research and Training Institute, Ministry of Internal Affairs and Communications, Japan

Table II
EXAMPLE OF CATEGORY OF TIME UNIT PRICE AND ITS AREAS

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| hourly pays | 700 | 700∼800 | 800∼900 | 900∼1k | 1k∼1.1k | 1.1k∼1.2k | 1.2∼1.3 |
| daily pays | 6k | 6∼7k | 7∼8k | 8∼9k | 9∼10k | 10∼11k | 11∼12k |
| monthly pays | 170k | 17∼180k | 180∼190k | 190∼200k | 200∼210k | 210∼220k | 220∼230k |
| annual incomes | 1.5m | 1.5∼2m | 2∼2.5m | 2.5∼3m | 3∼3.5m | 3.5∼4m | 4∼4.5m |

**Job Title**: The representation of Job title also follows the classification defined by the SB-MIC. Note that the depth of the tree is 3.

**Place of Business**: When calculating the similarity of "place of business" attribute (SPB), we consider the following two aspects from the points of view of job seekers: (1) Hometown Preferred Type and (2) Urban Area Preferred Type. When a user is a hometown preferred type, the user may have in mind of a special area or adjacent areas of the area to work. In this case, the value of SPB becomes a greater value as two places, target place and candidate place, are closer. We call this SPB value, SPB-HP value. When a user is an urban area preferred type, the user attaches importance to living environments. In this case, the value of SPB becomes a greater value as the economic level of the two places are closer. We call this SPB value, SPB-UP value. Considering these above assumptions, SPB is defined as the average of SPB-HP and SPB-UP. When calculating the SPB-HP, we classify 47 prefectures in Japan as a hierarchical tree structure of 3 depth. For example, Tokyo is a leaf node of the tree; the parent node of Tokyo is South KANTO; the parent node of South KANTO is KANTO; the parent node of KANTO is Japan, the root of the tree. Now we define SPB-HP($a_i$, $a_j$) as a function, which returns the value of SPB-HP between a target node $a_i$ and a candidate node $a_j$.

In order to compute the value of SPB-UP, we consider the prefecture order of gross product[4] and define the SPB-UP as follows: SPB-UP($tp$, $cp$)=$1 - \frac{2*diff(tp,cp)}{\#P}$

$diff(tp, cp)$ returns the difference of orders between target place $tp$ and candidate place $cp$. $\#P$ is the number of prefectures in Japan, 47.

**Academic Background**: According to an admission information site in Japan, the representation of Educational Categories of college or university consists of the hierarchical tree structure with the depth of 4. In addition, we consider the academic degrees, which consist of 8 categories: PhD, Master, Bachelor, Junior College, Technical School, High School, and Others. If the category of academic degrees between a target and a candidate node is different, the similarity of the two is 0.

**Salary**: is classified by the time unit price (TUP) of job, such as hourly, daily, weekly, or monthly pays, and annual incomes, and is provided price areas for each TUP class. When attribute $a_i$ and $a_j$ of Salary is in the same TUP class, if $a_j$ is the same as or upper area of $a_i$, $Sim_{Attr}(a_i, a_j)$ is 1; if $a_j$ is in one lower area of $a_i$, $Sim_{Attr}(a_i, a_j)$ is 0.5; otherwise 0.

Table II shows an example of each category of time unit price.

**Work Hours**: The Work Hour consists of a pair of start and end time of duty. The similarity between a target $a_i$ and a candidate work hour $a_j$ is computed by the ratio of hours overlapped between the two, as follows:

$$Sim_{Attr}(a_i, a_j) = \frac{2 * (a_i \cap a_j)}{|a_i| + |a_j|} \quad (5)$$

Note that $a_i \cap a_j$ is hours overlapped between $a_i$ and $a_j$ and $|a_i|$ stands for the hours, the length of $a_i$.

**Employment Pattern**: The employment pattern consists of 5 categories: Permanent Position, Contract Employee, Temporary Employee, Part-time Jobber, and Part-timer. The similarity is computed by exact-match base, i.e, the similarity is 1 if both a target node and a candidate are exactly matched, 0 otherwise.

### C. Popularity Degree

We assume that as a user gets a lot of attention of other users, the user becomes more popular, and that the value of popularity degree of the user increases as the popularity of other user who pays attentions to the user is greater. Considering this assumption, we regard that if user $A$ is chosen by user $B$ in case of any actions related to matching such as search, applying, scout, or positive reply, the user $A$ gets attention from the user $B$.

To compute the popularity degree of each user, we make a square matrix whose elements consist of the initial values of evaluation degrees between users, and get the eigen vector of the users with power method. The eigen value of a user corresponds to the popularity degree of the user. The initial value of popularity degree of each user is defined as follows[5]:

1) User with Job History: A user who quit a job and became a job seeker, gives an evaluation degree to the job, which corresponds to the value of a recruiter of the job, and at the same token, the recruiter gives an evaluation degree to the user. The initial value of

[4] http://www2.ttcn.ne.jp/honkawa/4550.html

[5] The simulation described later considers only the case of "User without Job History".

the popularity degree of a user is the average of the evaluation degrees given by other users.

2) User without Job History or without any evaluation degrees: The default value is assigned to a user as the initial value of the popularity degree of the user.

### D. Evaluation Degree

If user $A$ selects user $B$ in conducting matching actions such as search, applying or scout, user $A$ gives user $A$'s popularity degree to user $B$ according to the order of user $B$ in user $A$'s ranking list. User $B$ also gives user $B$'s updated popularity degree to a candidate user chosen by user $B$ according to the ranking list of user $B$. The popularity degree converged after these operations will be made an adjustment so that the total popularity degrees of users become equal to the initial total popularity degrees of the users. This idea is very similar to citation ranking of papers or Web pages (e.g. [6]). We call the degree given by user $A$ to $B$, **evaluation degree** from user $A$ to $B$. Function $Eval(A, B)$ in the equation of SHeU computes the evaluation degree from $A$ to $B$.

### E. Bi-directional Feedback

During matching process, when user $x$ takes action to user $y$ and user $y$ evaluates the action of $x$, these actions are regarded as bi-directional feedback for both $x$ and $y$. We put this bi-directional feedback computation in function $Eval$ defined in equation SHeU. To be more precise, function $Eval$ is computed based on the sum of $Update[act]$, which is an updated value corresponding to action $act$.

$Update[act]$ returns a positive value $\gamma$ if $act$ taken by a user is included in the positive actions such as search, applying, question, scout, accept of the actions offered by the other users. $Update[act]$ returns a negative value $-\gamma$ if $act$ is included in negative actions such as reject of the actions offered by the other users, or unsuccess of interview. We set $\gamma$ to 1 for experiments to be mentioned in Section III.

In addition, $Update[act]$ value between user $x$ and $y$ is propagated to user $i$ and $j$ who are similar to $x$ and $y$, respectively based on SHoU described in Section II-A.

*Propagating of Feedback:* When user $x$ took action $act$ to $y$, and received an evaluation from $y$, evaluation degree $Eval(x, y)$ from $x$ to $y$ is recomputed. $Eval(i, j)$ is an $i$-$j$ element of evaluation matrix of all users. The matrix is used for calculating popularity degree of users as mentioned in Section II-C. Thus, when $Eval(x, y)$ value is changed after user $x$ takes some action to $y$, popularity degree of users are recomputed. In the same way, the evaluation degrees $Eval(i, j)$ of users who are similar to $x$, say $i$, and $y$, say $j$, are also recomputed with Feedback Propagation $FP[act](i, j)$, which is a weight of $Eval(i, j)$. $FP[act](i, j)$ is defined as follows:

$$FP[act](i,j) \leftarrow FP[act](i,j) + Sim_{ho}(x,i) * Sim_{ho}(y,j) * Update[act] \quad (6)$$

Where $\leftarrow$ substitutes the value of equation in the right-hand side for that in the left-hand side. We assume that user $x$ and $y$ are heterogeneous users, $x$ takes action $act$ such as $applying$, $scout$, or $interview$ to $y$, and $y$ evaluates $act$ of $x$. User $x$ and $i$, and user $y$ and $j$ are homogeneous users, respectively. $Sim_{ho}(x, i)$ and $Sim_{ho}(y, j)$ are SHoU for $x$ and $i$, and for $y$ and $j$, respectively.

## III. SIMULATION

Showing the promising results with simulation is required to convince stakeholders such as developers, operators, or host of job matching systems so that they can provide us the real data of job matching systems. In this section, we describe the way to evaluate the proposed method with simulation and discuss the simulation results.

### A. Preliminary

Conditions of job seekers are generated by selecting some of recruiters' attributes. Each attribute in the conditions is selected with 50% of probability. Those of recruiters are generated in the same way. The attributes of job seeker consist of academic background and job history, and those of recruiters are place of business, salary, job title, category of business, work hour and employment pattern. The values of each attribute of both users are distributed normally with the average and variance values considering the areas predefined for the attribute. An example of the areas for Salary can be seen in Table II.

We also introduce two attributes that are not appeared in profiles and conditions: communication competency and job history. The value of communication competency is set to a randomly selected value between 0 and 1. The communication competency attribute is used as such the attribute that is made known in the interview. A value of a job history attribute of a job seeker is set by selecting a few numbers of recruiters. The recruiters in the job history are eliminated from the ranking list of the job seeker's candidates. The number of job seekers accepted by a recruiter is set to 1 for simplicity. So, the matching between a job seeker and a recruiter is one to one in this simulation.

### B. Matching Process Flow

A matching process flow of job seekers and recruiters consists of the following 7 steps, from (1) to (7). We call one matching flow, one term. When a user is rejected by other role's users who the user applied to at the actions of application, scout, or interview request, and the user can not move to the next step in the way of the term, the user has to wait for the next term; the next term will not start until no user conducts any activities in the current term. This time setting means job matching activities of every user move in every term in parallel. All users conduct their matching activities during their matching process until they become successful in their matching activity or the number of their trials goes over the limit of the terms, which is 5 in this simulation.

(1) Job seekers and recruiters create their ranking list of matching candidates, respectively.

(2) A job seeker applies to the top $n$ candidates, where $n = 2$. When a recruiter does not receive applications from top $n$ candidates of the recruiter, the recruiter scout the candidates which did not apply to the recruiter. After these applying and scout actions, the popularity and evaluation degree of the users are recomputed based on the actions.

(3) The users judge if they can move on to the interview of the other users who apply to or scout them in step (2). The criteria of judgment that they move on to the interview is whether or not the users who gave the offer by applying or scout are in the top half of their ranking list. If the users are in, they accept the users' offer; otherwise, the offer is rejected.

(a) go to step (4) if they move on to the interview.

(b) go to step (1) and wait until the next term get started, otherwise.

(4) In the interview, the recruiter evaluates the job seeker mainly from the view points of job seeker's communication competency. If the job seeker is in the top $\frac{N}{m}$ of the recruiter's ranking list, the job seeker passes the interview. Otherwise, the job seeker is rejected. Where $N$ stands for the number of candidates and 5 different numbers of $N$: 20, 50, 100, 150, and 200 are used in this simulation. $m$ is set to 3.

(5) A job seeker accepts the notification of acceptance to the interview if the recruiter is in the top $n$ of the job seeker's ranking list. Otherwise, the job seeker rejects the notification. After the notification actions, the popularity and evaluation degree of the users are recomputed based on the actions.

(6) The recruiter sends the notification of appointment to the top $k$ job seekers who accepted the recruiter's notifications of the interview. Other job seekers, who accepted the recruiter's notifications of the interview, but are not in the top $k$ of recruiter's candidate list, are put in the reserved list.

(7) The job seeker accepts the top of the recruiter who sent the notification of appointment to the job seeker, rejects other recruiters, and finish off the job hunting. The recruiter who achieves the $k$ job seekers also finish off the recruitment activities.

In the step (6) and (7), $k$ is set to 1.

### C. Evaluation Method

In the simulation, we compare the proposed method with the following two methods: a method without feedback (MwoF) and a method without propagation of feedback (MwoPF). The difference between the proposed method and the MwoF method is that the MwoF does not use bidirectional feedback information based on users' matching activities; The MwoF creates a ranking list of candidates of user $X$ by SCP, but does not change the list during the matching process. The difference with the MwoPF method is that the MwoPF uses the bidirectional feedback information, but does not propagate the feedback to similar users.

The evaluation measures are matching cost, which is the total number of actions taken by a user during matching process of the user, and matching score considering the matched users' ranking.

1) **Matching Cost**: is the total number of actions during the matching process of a user; the matching process of the user gets started by search action of matching candidates and ends when the user finds a matched user or the number of terms of the user goes over the limit. We assume the cost of any actions is equally 1 step although the cost of interview would be greater than that of search or apply in real situations. The precise value of each action's cost will be considered in our future work.
2) **Matching Score ($MatS$)**: $MatS$ between a job seeker $s_i$ and recruiter $r_j$ is defined as follows:
$$MatS(s_j, r_j) = \frac{1}{rank(s_i(r_j)) \cdot rank(r_j(s_i))} \qquad (7)$$
$rank(x(y))$ is the rank of $x$ to $y$. The order of the rank is at the time of matching. In the case of the approach without feedback, the rank order in the ranking list does not change during the matching process. On the other hand, in the case of the approach with feedback, the rank order of matching candidates in the ranking list may change according to the results of user actions. This means the rank order of a matching candidate at the end of the matching may be different from that at the starting time of the matching.

We use 5 different numbers of both job seekers and recruiters: 20, 50, 100, 150 and 200. After repeating the simulation 15 times by changing seeds for random number generation, we compute the average value of each parameter.

### D. Discussions on Simulation Results

Figure 1 and 2 say that the matching cost of the proposed method is the smallest among threes, and the number of matching of the proposed method also the greatest. These contribute the success of recommendation for matching, in other words, the success of the reconstruction of ranking of matching candidates.

Especially, we can say that the key factor to the effect is the propagation of the bi-directional feedback, as shown in Figure 2. Although the propagation of feedback is currently conducted to all the users according to their similarity, we need to reduce the target users of propagation for speeding up. Moreover, it has possibilities to improve the success rate

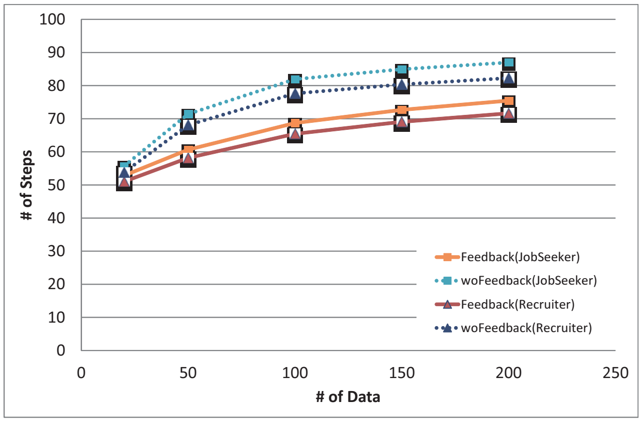


Figure 1. Matching Cost

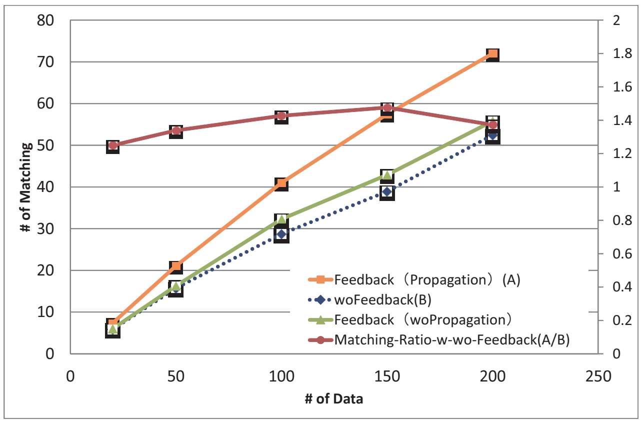


Figure 2. The Number of Matching

of matching and matching score since the propagation to all the users may lead in part to the bad effects.

As shown in Figure 3, the average matching scores of the three are almost equal except for the case of the number of data is 20. This is because all users take precedence of matching with the other users at the top of their candidates

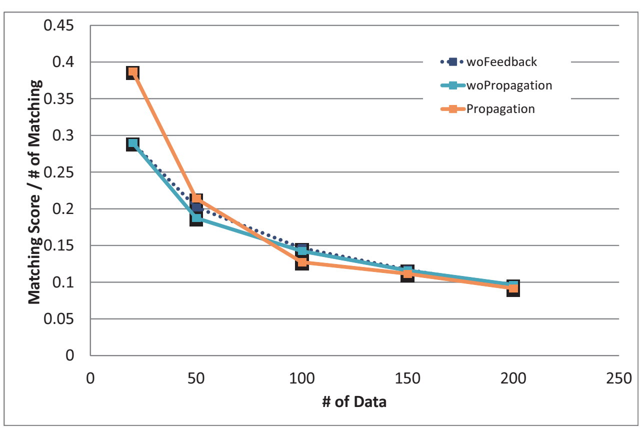


Figure 3. Rate of Matching Score to the Number of Matching

in all methods. More precisely, since the limit of trials, the number of terms is restricted to 5, and all users apply to the top two candidates in each turn, they can try their matching with the top 10 in their candidate ranking list and the candidates who applied to them and are in the top half. The MwoF method does not change the ranking list of the users, and the MwoPF method can change it, but the number of opportunities to apply is smaller than that of the proposed method. If users' candidates to be matched are in the lower rank than the top 10 in their original candidate list, they have little chance to meet with each other in those methods. On the other hand, since the proposed method change the ranking of their candidates, their matching chance increases. This is because the proposed method takes the greatest number of success of matching among three methods.

## IV. Conclusion

This paper proposed and discussed a reciprocal recommendation method with bi-directional feedback for job matching. In order to confirm the the validity of the proposed method, we conducted a simulation to compare the proposed method with the other two methods: a method without feedback (MwoF) and a method without propagation of feedback (MwoPF). The simulation results said that the proposed method had the best performance on the matching cost and success rate, and all the three methods had almost the same score on the average matching score.

Since this research has just started, there are a lot of things to do, especially we have to clarify which information can be gathered in real situations during job matching process and which information is significant to recommendation for job matching. To answer these questions, we are now preparing to gather the real job matching data, cooperating with a company that runs job matching support systems. Therefore our future work is to confirm the effectiveness of our method using the real data.

## References


[1] D. Gale and L. Shapley, "College admissions and the stability of marriage." *American Mathematical Monthly*, vol. 69, no. 1, pp. 9–15, 1962.

[2] L. Pizzato, T. Rej, T. Chung, I. Koprinska, and J. Kay, "Recon: A reciprocal recommender for online dating," in *RecSys2010*, 2010.

[3] J. Akehurst, I. Koprinska, K. Yacef, L. Pizzato, J. Kay, and T. Rej, "Ccr - a content-collaborative reciprocal recommender for online dating," in *22nd International Joint Conference on Artificial Intelligence*, 2011, pp. 2199–2204.

[4] F. Diaz, D. Metzler, and S. Amer-Yahia, "Relevance and ranking in online dating systems," in *SIGIR2010*, 2010, pp. 66–73.

[5] J. Malinowski, J. Malinowski, O. Wendt, and T. Weitzel, "Matching people and jobs: A bilateral recommendation approach," in *the 39th Hawaii International Conference on System Sciences*, 2006.

[6] L. Page, S. Brin, R. Motwani, and T. Winograd, "The pagerank citation ranking: Bringing order to the web." Stanford InfoLab, Technical Report 1999-66, November 1999, previous number = SIDL-WP-1999-0120. [Online]. Available: http://ilpubs.stanford.edu:8090/422/