

A Recommender System for Job Seeking and Recruiting Website

Yao Lu
React Group, EPFL
Lausanne, Switzerland
+41216936876
yao.lu@epfl.ch

Sandy El Helou
React Group, EPFL
Lausanne, Switzerland
+41216936873
sandy.elhelou@gmail.com

Denis Gillet
React Group, EPFL
Lausanne, Switzerland
+41216935168
denis.gillet@epfl.ch

ABSTRACT

In this paper, a hybrid recommender system for job seeking and recruiting websites is presented. The various interaction features designed on the website help the users organize the resources they need as well as express their interest. The hybrid recommender system exploits the job and user profiles and the actions undertaken by users in order to generate personalized recommendations of candidates and jobs. The data collected from the website is modeled using a directed, weighted, and multi-relational graph, and the 3A ranking algorithm is exploited to rank items according to their relevance to the target user. A preliminary evaluation is conducted based on simulated data and production data from a job hunting website in Switzerland.

Categories and Subject Descriptors

H.3.5 [Information Storage and Retrieval]: Online Information Services – *Web-based services*; H.3.1 [Content Analysis and Indexing] – *Indexing methods*.

General Terms

Algorithms, Design.

Keywords

Job seeking and recruiting websites, recommender system, multi-relational graph.

1. INTRODUCTION

In this paper, we present a recommender system designed for job seeking and recruiting websites. The recommender system will be deployed on a website developed to provide an European career center network, helping companies in recruiting top talents in shaping their individual career paths. The details of the website are omitted in the paper due to a confidentiality agreement. The website does not only allow users to create their profiles but also support rich interaction features to apply for a job or mark ‘like’ to an employer. The proposed recommender system aims at leveraging the jobs and companies that are deemed important for a target candidate and vice versa. To meet this objective, user profiles, such as job descriptions and candidate CVs are examined along different forms of user actions. The recommendation approach is modeled on a directed weighted graph, where the

content-based and interaction-based relations are translated into edges connecting different entities (candidates, employers and jobs). Then, the entities are ranked according to their relevance to the target users, considering the preferences of both employers and candidates. The rest of the paper is organized as follows: section 2 discusses related work; section 3 introduces graph modeling and recommendation scenarios; section 4 introduces the hybrid recommendation approach applied in the system; section 5 discusses experiment results; and section 6 concludes the paper.

2. RELATED WORK

Recently, job seeking and recruiting websites have been experiencing a striking rise [1]. As the amount of information grows, a recommendation system is necessary to help match the right candidate with the right job. To do so, recommendation techniques such as content-based filtering, collaborative filtering and hybrid approaches can be applied [2]. The content-based approach matches candidate profiles with employer profiles and job requirements. Initially based on keyword search, content-based filtering was improved into a statistical inference and semantic engine to match profiles rather than keyword [3]. Besides, previous studies state that the challenge of matching candidates and jobs is grounded in the interactionist theory of behavior [4] and believe that interactions are important for recommendation [5] as they strongly influence the candidate’s job choice and employer’s hiring decision. Some interaction-based recommendation systems, such as CASPER [6], make use of collaborative filtering to recommend jobs to users based on what similar users have previously liked. Hybrid systems are also exploited to match people and jobs. A hybrid system proposed by Malinowski is based on the idea that a good match between people and jobs needs to consider both the preferences of the recruiters and the candidates. As a result, the matches between jobs and candidates are predicted according to candidate CVs and employer descriptions and requirements as well as previous rating information [7]. The limitation of this method is that it is difficult for the users to rate the jobs they have not worked on yet. To the best of our knowledge, no previous published work has applied a recommender system integrating both content and user actions in the job seeking and recruiting process.

3. GRAPH MODELING AND RECOMMENDATION SCENARIOS

In our system, each entity has a profile. We use the content- and interaction-based relations to connect entities as follows:

Content-based relations:

Profile match: we build this bidirectional relation based on the hypothesis that if a candidate’s CV content matches a job’s description, they are probably interested in each other.

Profile similar: two entities of the same group are connected with this bidirectional relation if they have similar profiles. This is built on the hypothesis that the users with similar profiles tend to have the similar interests and opportunities.

Interaction-based relations:

Post: we build a bidirectional relation between the employer and its jobs. The target users of the website are students or people who just graduated, for whom the employers are rather important. When a user expresses interest in an employer, his/her interest is also delivered to the jobs posted by the employer. Thus, when comparing two similar jobs posted by different employers, the one from the preferred employer gains more importance. Besides, if a user shows his/her preference to many jobs from the same employer, this employer gains more importance.

Apply: when a candidate applies for a job, we build a single directed relation, *Apply*, from the candidate to the job. We believe this is an explicit signal that the candidate is interested in the job. This signal can lead the candidate to other jobs similar to the ones he/she applied for.

Favorite: a user can add an entity into his/her ‘favorite list’. This is also a strong and explicit signal of interest. Thus, a relation from the user to his/her favorite entity is built.

Like: similar to Facebook and other prominent social media sites, there is a ‘like’ button on the entity’s page. The difference between ‘like’ and ‘add to favorite’ or ‘apply’ is that the user may not revisit the items they liked. For example, the job is expired or the candidate likes the task but he/she does not want to work in the place offered by the job, the user may click ‘like’ instead of ‘apply’. We build such a single directed relation from the user to the liked item because it leverages his/her interests. The ‘like’ action is considered as an explicit feedback but weaker than ‘apply’ and ‘favorite’.

Visit: this relation is built from the user to the item he/she visited. ‘Visit’ is considered as implicit feedback because there is a probability that the user does not show any interest in the item after having read its description and other details.

Table 1. Inter-relations of Entities

User Object	Candidate	Employer	Job
Candidate	Similar	Visit, Like Match Favor Apply	Visit, Like Match Favor Apply
Employer	Visit, Favorite Match	Similar, Visit	Post, Visit
Job	Match	Posted	Similar

Table 1 shows the entities and their bilateral relations. Based on that, a weighted multi-relational graph is modeled. Its nodes represent the entities and its edges represent the inter-relations listed above. The importance weights of the relations are set as follows:

$apply > favorite > post = like > similar = match = visit$

In our system, there are four recommendation cases: recommending employers to candidates, jobs to candidates,

candidates to employers, and candidates to jobs. The entities of the same group are not recommended to each other. More specifically, we introduce some recommendation scenarios on the simplified illustrating graph in Figure 1.

1. Recommend Job C to Peter because Yao liked Job C, and Peter and Yao have similar profiles;
2. Recommend Job A to Peter because Peter applied for Job B, and Job A and Job B are similar;
3. Recommend Employer 3 to Peter because Peter applied for Job B and Job B is posted by Employer 3;
4. Recommend Job B to Yao because Peter applied for Job B, and Peter and Yao have similar profiles.

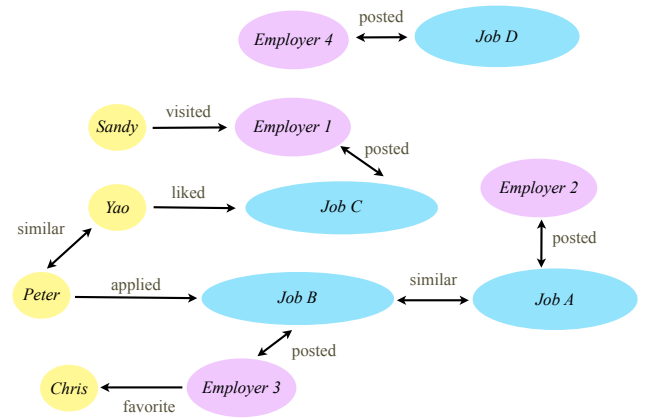


Figure 1 Simplified Illustrating Graph

4. HYBRID RECOMMENDATION

A pipelined hybrid recommendation approach is implemented in this system, where the result of content-based similarity is fed into a relation-based algorithm as an additional relation after normalization. Figure 2 shows the recommendation framework. After the graph is constructed as described in section 3, a target user receives a personalized recommendation list extracted from the graph via importance computation approaches.

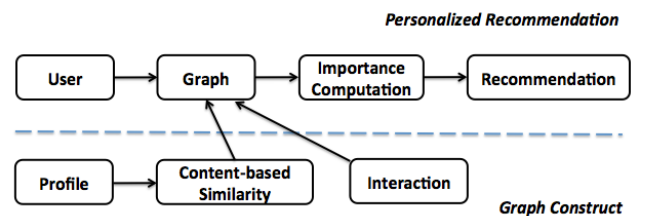


Figure 2 Recommendation Framework

4.1 Content-based Similarity Computing

We build content-based relations by comparing the profiles of two entities. When computing the *Profile Similar* relation, we compare two entities from the same group with their corresponding fields

of their online profiles. We select the following fields that are considered as important to characterize an entity and compute profile-based similarity:

Candidate: age, gender, interest, language, education (diploma, major, university), skills, work experience, self-description;

Employer: country, industry field, scale, description;

Job: employer, location, requirement, task, opportunity.

The similarity between two entities of the same type is the weighted sum of similarity values of each field. The main challenge lies in setting the field weights. Since our target candidate users are mainly students, we set ‘interest’ and ‘education’ as key factors to candidates. A previous study of the profile patterns of the website [8] shows that the employer industry field and job company are key factors to characterize an employer and a job respectively.

In order to compute profile similarities between jobs and candidates, files are parsed into plain text and similarity is computed using the Latent Semantic Analysis (LSA) tools.

The top 1% most similar pairs are connected with *Profile Match* or *Profile Similar* relation. Computing content-based similarity computing helps overcome the cold-start problem caused by interaction sparsity. Despite the advantages of content-based similarity measures, we have many challenges to achieve good results by applying the content-based method exclusively, for example, it requires a lot of work such as profile formatting, job/employer title canonicalization, and creating industry taxonomy. As a result, interaction-based relations are built from user logs and translated into graph edges.

4.2 Ranking Algorithm

4.2.1 Personalized Multi-Relational PageRank

The key idea of the original PageRank is that if the owner of a page j links to a page i , the owner is implicitly indicating that page i is important [9]. Thus, every node in the graph earns its importance from its incoming edges and delivers importance to others nodes via its outgoing edges. The 3A ranking algorithm [10] is a personalized and contextual ranking method extended from PageRank. The algorithm is applied in our system to extract the rank vector from the constructed graph and it is reformulated to fit the job/employer and candidates matching:

An entity is relatively important to a target entity, if and only if many important entities of the target entity delivered interests or have similarities to it via important content- or interaction-based relations.

According to the 3A ranking algorithm, the importance matrix M can be written as follows:

$$M = \frac{\lambda}{N} I + d \sum_{e \in E} w_e (T^e + D^e) + p_u U \quad (1)$$

where $\sum_{e \in E} w_e = 1$; $\lambda + d + p_u = 1$; $d > 0$, $p_u > 0$ $\lambda > 0$

In Equ. 1, $T^e(N \times N)$ is the transition matrix for relation e , $D^e(N \times N)$ is the dangling node matrix for relation e . $I(N \times N)$ is a matrix where all the elements are 1. And w_e represents the weight of relation e ; d is damping factor representing the probability of going from one node to the others by following different relation types; p_u is the personalization factor used to boost the

importance of entities connected to the target; λ is the probability to jump to random nodes and $U(N \times N)$ is the matrix where all the rows are 0 except for the one corresponding to the target entity. The rank vector R of a target node is calculated iteratively using Equ. 2. More details regarding the convergence and computation of the rank vector can be found in the literature [9].

$$R^{k+1} = MR^k \quad (2)$$

4.2.2 Personalization Factor for Implementation

The personalization factor boosts the score of the target user’s neighbors. Direct neighbors are those directly connected to the user and derived from his/her previous actions. Indirect neighbors are those connected through a *Profile Match* or intermediate nodes. Since the user is aware of the direct neighbors, they are removed from the final recommendation list. Figure 3 shows the ratio of second-degree neighbors to the total number of relevant items recommended. The graph shows that the bigger the personalization factor is, the more relevant items will be recommended, and the closer they will be to the target user (the second-degree neighborhood ratio increases with the personalization factor). This means, that a trade-off should be set so that relevant but distant items are recommended. We empirically chose the personalization factor to be 0.6, taking into account our dataset and literature recommendations.

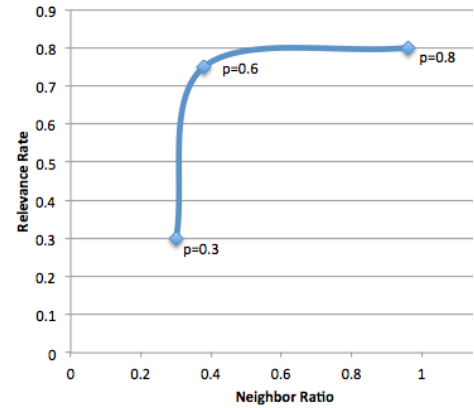


Figure 3 Personalization Factor Effect

5. EXPERIMENTS

As a proof of concept, the ranking algorithm is firstly computed on the sample graph in Figure 1. Ranked lists for candidate Peter and Job B are shown in Table 2. For Peter, Job B has the highest score since Peter applied for the job. As expected, Job A ranks higher than Job C because the importance weight of *applying* for a job is higher than *liking* a job. For Job B, Chris ranks first because Employer 3 *favoured* Chris. Results for the other entities also show that the recommendation result went with the scenarios introduced in section 3.

The production dataset contains 7000 candidates, 400 employers and 8000 jobs. Our approach is compared to pure content-based profile similarity (PS) and collaborative filtering (CF) in terms of precision and user coverage, using the website data at hand. With respect to the user coverage, PS is able to cover 37% of website users who provided well-completed profiles and user coverage of

CF is quite low because only 9% of the users had interactions. Our hybrid recommender system provides recommendations to the union of the users covered by CF and PS, that is, 39.6% of all the users.

With respect to precision measures, since interaction features have only been added recently to the site, user actions are still very sparse. Hence we used all the data at hand to compute recommendations and manually evaluated the recommendation quality for 9 entities as targets. The relevance rate of the top 10 items in each list is verified manually. Table 3 shows the precision of our hybrid system, PS and CF. CF is unavailable when we recommend candidates to jobs (C to J) because the jobs don't deliver interactions to others. For every recommendation list, the highest precision is marked in bold and italic. On average, our hybrid recommender system outperforms PS ($p = 0.005$) and CF ($p = 0.02$).

Table 2. Ranked Lists for Peter and Job B

Target: Peter		Target: Job B	
Job B	0.336161117	Job A	0.331817212
Yao	0.331842704	Employer 3	0.331322184
Job A	0.036523597	Job B	0.043462022
Employer 3	0.036077245	Chris	0.039990737
Job C	0.034145923	Employer 2	0.035359156
Peter	0.033693888	Employer 1	0.031582942
Chris	0.032381152	Job C	0.031373904
Employer 1	0.032288226	Job D	0.031149813
Employer 2	0.03188232	Employer 4	0.031149813
Employer 4	0.031816419	Yao	0.03114676
Job D	0.031816419	Peter	0.030931743
Sandy	0.031370989	Sandy	0.030713715

Table 3 Precision of Top 10 Recommendation Results between Employers (E), Candidates (C) and Jobs (C)

	Target	Hybrid	PS	CF
E to C	Candidate 1	0.8	0.1	0.3
	Candidate 2	0.5	0	0.2
	Candidate 3	0.1	0	0.2
J to C	Candidate 1	0.7	0.3	0.5
	Candidate 2	0.5	0.4	0.4
	Candidate 3	0.5	0.3	0.4
C to E	Employer 1	0.5	0.2	0.2
	Employer 2	0.8	0.8	0.6
	Employer 3	0.7	0.2	0.5
C to J	Job 1	0.7	0.3	N/A
	Job 2	0.1	0	N/A
	Job 3	0.2	0.8	N/A

6. CONCLUSION

In this paper, we presented a hybrid, personalized recommender system for job seeking and recruiting websites. The interaction

features are designed along with the recommender system. A graph based ranking algorithm is taking into account the business logic in the job seeking and recruiting process. A preliminary evaluation based on dataset from production website shows that our system outperforms content-based profile match and collaborative filtering on recommendation precision and user coverage. In the future work, user studies and evaluations based on online data will be conducted to evaluate the accuracy, usability and other characteristics of the proposed recommender system and refine it accordingly.

7. REFERENCES

- [1] Jobvite 2011 Social Recruiting Survey. Dec. 2011. <http://recruiting.jobvite.com/>
- [2] Balabanovic, M. and Shoham, Y. 1997. Fab: Content-based, collaborative recommendation. *Communications of the ACM*, 40, 3 (Mar. 1997), 66-72. DOI = 10.1145/245108.245124.
- [3] Sovren Group. 2006. *Overview of the Sovren Semantic Matching Engine And Comparison to Traditional Keyword Search Engines*. Sovren Group, Inc.
- [4] Sekiguchi, T. 2004. Person-organization fit and person-job fit in employee selection: a review of the literature. *Osaka Keidai Ronshu* 54, 6 (2004) 179-196.
- [5] Malinowski, J., Weitzelb, T., Keima, T. 2008. Decision support for team staffing: An automated relational recommendation approach. *Decision Support Systems*, 45,3 (June. 2008), 429 - 447. DOI = 10.1016/j.dss.2007.05.005.
- [6] Rafter, R., Bradley, K., and Smyth, B. 2004. Automated collaborative filtering applications for online recruitment services. In *Proceedings of the International Conference on Adaptive Hypermedia and Adaptive Web- based Systems*. 1892, (Trento, Italy, August. 2000). AH '00, 000, 363-368. DOI: 10.1007/3-540-44595-1_48.
- [7] Malinowski, J., Keim, T., Wendt, O., Weitzel, T. 2006. Matching People and Jobs: A Bilateral Recommendation Approach. In *Proceedings of the 39th Annual Hawaii International Conference on System Sciences*. 6 (Hawaii, USA, Jan. 4-7, 2006). HICSS '06, 137c.
- [8] Lu, Y., Helou, S. E., Gillet, D. 2012. Analyzing User Patterns to Derive Design Guidelines for Job Seeking and Recruiting Website. In *Proceeding of The Fourth International Conferences on Pervasive Patterns and Applications*. (Nice, France, July. 23-27, 2012) Patterns 2012. IARIA, 11-16.
- [9] Page, L., Brin, S., Motwani, R., Winograd, T. 1998. *The PageRank citation ranking: Bringing order to the Web*. Technical report. Stanford Digital Library Technologies Project.
- [10] Helou, S. E., Salzmann, C., Gillet, D. 2010. The 3A Personalized, Contextual and Relation-based Recommender System. *Universal Computer Science*, 16, 6 (Aug. 2010), 2179-2195. DOI = 10.3217/jucs-016-16-217