Recommending Collaborators Using Keywords*

Sara Cohen
Dept. of Computer Science and Engineering
Hebrew University of Jerusalem
sara@cs.huji.ac.il

Lior Ebel
Dept. of Computer Science and Engineering
Hebrew University of Jerusalem
lior.ebel@mail.huji.ac.il

ABSTRACT

This paper studies the problem of recommending collaborators in a social network, given a set of keywords. Formally, given a query q, consisting of a researcher s (who is a member of a social network) and a set of keywords k (e.g., an article name or topic of future work), the *collaborator recommendation problem* is to return a high-quality ranked list of possible collaborators for s on the topic k. Extensive effort was expended to define ranking functions that take into consideration a variety of properties, including structural proximity to s, textual relevance to k, and importance. The effectiveness of our methods have been experimentally proven over two large subsets of the social network determined by DBLP co-authorship data. The results show that the ranking methods developed in this paper work well in practice.

Categories and Subject Descriptors

H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval

Keywords

collaborator recommendation; social network

1. INTRODUCTION

Context-based person recommendation is the problem of recommending people in a social network, when given a set of keywords k, by a member of the social network s. The goal is to return a high-quality ranked list of people who are relevant both to k and to s, and, perhaps, are of global importance. Instances of this problem include problems such as recommending an accessible expert on a medical issue, recommending a good lawyer for a legal entanglement, or (the problem on which we focus in this paper) recommending collaborators on a research topic. These examples have much in common—in particular, it is crucial that the recommended people be both relevant to the topic at hand k, and relevant to the user s (who will later consult with, or collaborate with, people in the result).

*The authors were partially supported by the Israel Science Foundation (Grant 143/09) and by the Ministry of Science and Technology (Grant 3-8710).

WWW 2013 Companion, May 13–17, 2013, Rio de Janeiro, Brazil.
ACM 978-1-4503-2038-2/13/05.

In our paper we focus on one particular flavor of context-based person recommendation, which we call *collaborator recommendation*. Given a researcher s and a set of keywords k, the goal is to return a high-quality ranked list of potential collaborators for s on the topic k. We chose to focus on this problem for two reasons: importance and evaluability, as discussed next.

Importance. One factor in the eventual success or failure of a new research endeavor is the quality of the research collaborators. This is even more pronounced for interdisciplinary efforts, that require a collection of differing expertise. Therefore, one of the first questions that a researcher often asks herself upon embarking on a new research problem is with whom to collaborate. We note that the need to find fruitful collaborations also arises due to grant regulations.

Currently, efforts to form collaborations are often ad-hoc, with researchers simply continuing to collaborate with past partners, or by chance meetings at conferences or other gatherings. The ability to effectively locate relevant collaborators for a research topic is important, and can have long-lasting results on the quality of the work. In fact, developing mechanisms for finding collaborators in scientific social networks was recently recognized as an important and open problem [12].

Evaluability. Obviously, it is not sufficient to develop ranking functions for the problem at hand. As interesting and intuitively appealing as these functions may be, "the proof of the pudding is in the eating." In other words, it is crucial to experimentally evaluate the ranking methods to determine their effectiveness. Collaborator recommendation is a fertile ground for extensive experimentation efforts, as there is a wealth of available data, which can be utilized for experimental evaluation. Precisely how this data is leveraged, is discussed later on in Section 4.

This paper is a significant first step towards solving the collaboration recommendation problem, by combining traditional techniques for structural link prediction in social networks (which are often used for people recommendation [2]), with textual relevancy and global importance metrics. Effective ranking functions have been carefully developed (and experimentally evaluated) so as to yield quality results. We note that our methodology has been implemented in our *CollRec* system, and can be used in practice. While the methods presented in this paper have been experimentally evaluated over a co-authorship network, the metrics con-

sidered in this paper are interesting and likely to be useful for the more general context-based person recommendation problem. Further experimentation and validation of our approach to other domains is left for future work.

Related Work. There are several research areas that are related to collaborator recommendation. In *expert search*, e.g., [4], the goal is to find global experts within a social network for specific topics. Ranking functions for expert search have focused on importance metrics, and ignored the distance of the experts from the user who initiates the search. This is quite different from collaborator recommendation, where a collaboration is more likely to be formed between people who are close in a social network. (We note that even for the classic expert search, ignoring s may be a drawback, as the experts recommended may not be accessible to s.) There has also been significant recent work on *social network search*, e.g., [11], where social relations are used to rank data items (or Web pages), but not people.

Another related problem is that of *link prediction* [2,8,13]. In the link prediction problem, the goal is to predict links that are likely to be added to the social network. This is useful for friend recommendation. Link prediction differs from collaborator recommendation in that the latter is context-based, i.e., different people will be recommended as potential collaborators for different sets of keywords k. Thus, ranking methods for link prediction are not necessarily optimal for collaborator recommendation, and indeed, our experimentation yields the unsurprising result that incorporating context-based ranking metrics improves collaborator recommendation. We do, however, use a similar methodology to that of link prediction in evaluating the quality of our ranking methods.

Finally, there have been a few efforts to tackle the problem of recommending collaborators [3, 7, 9]. In [7] and [9], a collaborator t is recommended for a person s based both on the social network, and on the past research of t and s (e.g., if they worked on similar topics). However, they do not provide the capability of finding a collaborator that is appropriate for a specific topic k. The CollabSeer system [3] also recommends collaborators for a given scientist based on the scientific co-authorship network. However, again, the user cannot provide a specific context (i.e., research topic) for the collaboration search.[1] We note, also that [3] uses very different evaluation metrics, and calculates the quality of their results based on a projected usefulness function. On the other hand, we use the ground truth of established collaborations to evaluate the quality of our methods. Thus, this paper is the first to present effective ranking methods for collaborator recommendation, based both on the user s, and on the context k.

2. COLLABORATOR RECOMMENDATION

While this paper studies the collaborator recommendation problem, our techniques can be used for the more general context-based person recommendation problem. Hence, we discuss the data model and the problem of interest in a more general setting.

[1]They do have a very limited topic-specific search, but this is for a small set of topics that are generated automatically by CollabSeer based on past research of s. This list is limited, and does not include topics that are new for s.

A *social network* is an undirected multigraph $G(V, E)$, where V is a set of nodes (representing people) and E is a set of edges (representing associations among people). Each node v has a textual *profile description* annotated as $profile(v)$. Each edge e is associated with three attributes: *(1)* a *textual description* of the association $label(e)$, *(2)* the *time* of the association $time(e)$, and *(3)* the *setting* of the association $setting(e)$.

In a co-authorship network, the nodes are authors, and edges represent joint publications. Thus, $profile(v)$ is a bag of words comprised of all words appearing on publication titles of the author, while $label(e)$, $time(e)$ and $setting(e)$ are the title of the publication, the date of publication and the publication venue of the collaboration e, respectively.

A *query* is a pair $q = (s, k)$ where s is a node in the social network (called the source) and k is a set of keywords. Intuitively, the *collaborator recommendation problem* is: Given a query (s, k), rank the nodes of the network with respect to their likelihood of forming a collaboration e with s (and possibly with additional nodes) that has a textual description matching k. More formally, our goal is to develop a score function $score(u, q)$ that, given a query q, associates a numerical value with u, such that if $score(u, q) > score(v, q)$, then u is more likely than v to form a collaboration with s described by k. Note that in the co-authorship network, k is the title (or topic) of a paper and collaborator recommendation is the problem of finding people most likely to collaborate with s on a paper entitled (or about) k.

3. SCORE FUNCTIONS

Developing an effective function $score(u, q)$ is a difficult problem, especially since there are many different factors that can effect ranking. For example, closeness of a node u to the source s within the graph influences the likelihood of s to collaborate with u (e.g., people often collaborate on new papers with past co-authors). In addition, the degree in which u is an expert in k is important. This can be measured using $profile(u)$. The number of past collaborations, and their recency can influence the score function. Finally, the importance of u in the network is also a contributing factor for $score(u, q)$.

Our score functions are based on structural proximity, textual relevancy, importance of the nodes in the network, or on combinations of these factors. Structural proximity functions considered have proven useful in the past for the related problem of link prediction [1, 8], (which is often used for friend recommendation, when there is no context of keywords). The textual relevancy functions considered are a new factor that is shown to be useful for collaborator recommendation, and do not play a role in the standard link prediction problem.

Score functions are normalized so as to return values in the range $[0, 1]$. Due to space limitations, we present only the underlying ideas of each score function, and omit discussion of normalization. Finally, recall that $score(u, q) > score(v, q)$ indicates that u is a better answer (and hence, would be ranked higher) than v.

Structural Proximity. We fix a query $q = (s, k)$. Structural-proximity score functions for q attempt to quantify how close s is to a given node u. We study several variants of the distance metric. Specifically, given a weight function w that assigns a positive value $w(e)$ to each edge e, we use

$dist_w(u, q)$ to denote the weighted distance[2] of u from s. We consider three weight functions for an edge (v, w): constant weight of 1, the reciprocal of the *number of collaborations* of v and w, and the logarithm of the number of *time* units that have passed since the most recent collaboration of v with w. These are called *dist*, *cdist*, and *tdist*, respectively.

In addition to distance, we also measure structural proximity using the Adamic/Adar [1,8] score function, in which the score for a node u is a function of the number of collaborators common to s and u. To be precise, let $N(z)$ be the set of all neighbors (i.e., past collaborators) of z. Then, $ad(u, q) = \sum_{z \in N(u) \cap N(s)} \frac{1}{\log |N(z)|}$.

Textual Relevancy. Textual-relevancy score functions for q attempt to quantify how well u is described by the keywords k, i.e., how likely it is for u to collaborate in the future about k. We consider two score functions. The first function, called *tfidf* uses a TF-IDF based function to determine textual relevancy of u, by using $profile(u)$.[3]

The second function, called *collab*, is more intricate than *tfidf* and was defined especially for this setting, to leverage a variety of attributes of collaborations. To compute the score function *collab*, we take a two-step approach. First, we calculate the expression $\Phi(u, q)$ defined as

$$\sum_{(u,v) \in E} text((u, v), k) \times \beta_{v \in N(s)} \times \gamma_{setting((u,v),s)} \times \frac{1}{\log age(u, v)}$$

where

- $text((u, v), k)$ is a TF-IDF based matching score of $label((u, v))$ to keywords k;
- $\beta_{v \in N(s)}$ is a number $\beta > 1$ if $v \in N(s)$, and 1 otherwise;
- $\gamma_{setting((u,v),s)}$ is a number $\gamma > 1$ if s has an edge with the same setting as (u, v), and 1 otherwise;
- $age(u, v)$ is the age of the collaboration (u, v) (i.e., the number of time units which passed since $time(u, v)$).

Intuitively, $\Phi(u, q)$ weights relevant textual collaborations by their recency and setting, and also gives greater weight to the relevant collaborations of u if neighbors of s were collaborators.

For the second step in computing *collab* we use the *unseen bigrams* approach [8] as follows. Let Φ_n be the top n nodes according to Φ. Then, $collab(u, q)$ is $\sum_{v \in \Phi_n \cap N(u)} \Phi(v, q)$. We note that *collab* is unique in that it examines the textual relevancy of edges, while taking into account their weights. As we show later, a combination of this function with *tfidf* and other known proximity-based functions yields a superior score function over the corpora tested.

Global Importance. We have also considered two measures of global importance of nodes, for use in score functions: *degree* (i.e., the importance of a node is locally determined by the number of neighbors she has) and *PageRank* (in which importance is determined by a global consideration of the network). Experimentally, we have found that importance-based score functions do not improve predictions of future collaborators. Hence, our default scoring function, described next, does not take importance into account. However, we note that node importance can be useful for recommendation (e.g., to find an important person who is well-related to a topic). Therefore, in the *CollRec* system that we developed, our user interface allows users to specifically include importance in ranking results.

A Combined Score Function. We considered a variety of approaches to combining score functions, e.g., using normalized Borda score combinations [5, 10], as well as taking top results according to one score and reranking according to another. After exhaustively considering different combinations of functions of subsets of the data, and performing cross-validation, we have empirically found a combined score function which has proven effective. This function, called *SoCScore*, is a weighted sum of the scores of two sets of top n nodes: *(1)* the top-n nodes obtained by taking the top-n results by *tdist* and then re-ranking by *tfidf*, and *(2)* the top n nodes obtained by taking the top-n results by *ad* and re-ranking by *collab*. Further discussion of other types of combinations are omitted due to lack of space.

4. EXPERIMENTAL EVALUATION

Testing the effectiveness of a recommendation system is difficult, as it requires user-feedback to determine relevancy of the results. We circumvent this problem by instead testing for effectiveness of our score functions in *predicting* actual collaborations. This method of evaluation follows from the intuition that if the system can predict actual collaborations, then its recommendations are likely to be of interest.

We used two subsets of the DBLP co-authorship network for our experiments, containing database and artificial intelligence collaborations. The database network contains over 100,000 nodes and 700,000 collaboration edges, representing over 130,000 publications, while the artificial intelligence network contains over 80,000 nodes and 500,000 collaboration edges, representing over 80,000 publications.

To test whether our score functions correctly predict future collaborations, we generated a set Q of 200 random queries by randomly choosing authors s and publication titles k, such that s published an article (with collaborators) entitled k in years 2009-2011 (inclusive). When processing a query, the system considers data only until the year when the article k was published, exclusive. We note that we only chose queries representing conference papers, as journal paper collaborations are easier to predict (since a journal paper was often published with the same authors and title previously at a conference). Due to the small world property observed in the DBLP co-authorship network [6], and for efficiency reasons, our score functions only consider nodes that are up to 3 hops away from the source s.

We note that in our experimentation, the approach taken is similar to that taken for experimentally evaluating scoring functions for the link prediction problem [8], i.e., freeze the network and predict new edges that will be added into the network at a future point in time. However, when studying link prediction, usually one takes the top-n most likely edges to be added to the network, and then checks how many of these predicted edges actually form later in the network. In our experimentation, we take random queries and test how well we predict collaborations for these queries. We note

[2]The weighted distance is the sum of weights on the lightest path from s to u.

[3]We used Lucene (`http://lucene.apache.org/`), and its default ranking function, to determine this score.

	all collaborators			*only new collaborators*		
	TopScore@10	*Recall@10*	*MRR*	*TopScore@10*	*Recall@10*	*MRR*
dist	62.5	48.1	0.349	10.0	7.14	0.046
cdist	62.5	47.63	0.443	12.5	8.91	0.065
tdist	66.0	55.15	0.444	14.5	11.47	0.064
ad	6.0	3.30	0.027	15.0	11.26	0.067
tfidf	38.0	27.57	0.235	10.0	7.64	0.054
collab	27.5	22.86	0.185	16.0	11.65	0.081
SoCScore	66.5	55.43	0.425	20.5	16.09	0.113

Table 1: Experimental evaluation of score functions

that the latter problem is more representative of collaborator recommendation (where an arbitrary user may pose a query), but is also much more difficult, as it is possible that no predicted collaborations for a random query are among the top-n most likely collaborations when considering all possible queries (as is done for classic link prediction).

We measured the quality of the results for a score function *score* on Q using three measures:

- *TopScore@n*: the percentage of queries q in Q for which $score(q)$ ranked a true collaborator within the top n. Note that this is the ground truth, observed by seeing the actual publication details.
- *Recall@n*: the percentage of true collaborators within the top n, out of total number of true collaborators for s (existing in the graph prior to the year of publication) for article k.
- *MRR*: the mean reciprocal rank, i.e., $\frac{1}{|Q|}\sum_{q=1}^{|Q|}\frac{1}{rank_q}$, where $rank_q$ is the index of the first true collaborator.

It turns out that it is significantly easier to predict that a node u will collaborate with s, if u has already collaborated with s in the past, than for new collaborators. Therefore, in our experimentation, we studied two variations of the problem: recommending (any type of) collaborators for a given query, and recommending new collaborators for a query. The results for these problems, over the database subset of DBLP, appear in Table 1. Due to space limitations, we do not present results for the artificial intelligence network. However, the results were similar. In our table, we also omit the results for the global importance measures, as they were very low.

When predicting any type of collaborator, the weighted distance *tdist* and *SoCScore* give superior results (with a slight advantage for *tdist* in the *MRR* measure). For the harder problem of predicting new collaborators, *SoCScore* clearly outperforms the other functions.

5. CONCLUSIONS AND FUTURE WORK

This paper presented effective scoring functions for collaborator recommendation. Our experimental evaluation shows that a mix of both structural proximity measures and textual relevancy gives superior results, and in particular, *SoCScore* function is effective in recommending collaborators.

As future work, we plan on evaluating how additional textual data, such as publication abstracts for the scientific coauthorship network, can improve the results. We also will consider applying more sophisticated text-based techniques, such as topic extraction for both the textual profiles and the input query. More importantly, we intend to study how effectively our methodology carries over to additional domains of the context-based person recommendation problem. Finally, we intend to include machine learning techniques to further improve the quality of functions which combine several scoring features, and so as to automate the ability to utilize our results over additional domains.

6. REFERENCES

[1] L. Adamic and E. Adar. Friends and neighbors on the web. *Social Networks*, 25:211–230, 2001.

[2] L. Backstrom and J. Leskovec. Supervised random walks: predicting and recommending links in social networks. In *Proceedings of the fourth ACM international conference on Web search and data mining*, pages 635–644. ACM, 2011.

[3] H. Chen, L. Gou, X. Zhang, and C. Giles. Collabseer: a search engine for collaboration discovery. In *JCDL*, 2011.

[4] R. D'Amore. Expertise community detection. In *SIGIR*, 2004.

[5] J. C. de Borda. *Memoire sur les Elections au Scrutin.* Histoire de l'Academie Royale des Sciences, 1781.

[6] E. Elmacioglu and D. Lee. On six degrees of separation in dblp-db and more. *SIGMOD Rec.*, 34(2):33–40, June 2005.

[7] D. Lee, P. Brusilovsky, and T. Schleyer. Recommending collaborators using social features and mesh terms. *JASIST*, 48(1):1–10, 2011.

[8] D. Liben-Nowell and J. Kleinberg. The link prediction problem for social networks. In *CIKM*, 2003.

[9] G. Lopes, M. Moro, L. Wives, and J. de Oliveira. Collaboration recommendation on academic social networks. *Advances in Conceptual Modeling–Applications and Challenges*, pages 190–199, 2010.

[10] M. Renda and U. Straccia. Web metasearch: rank vs. score based rank aggregation methods. In *SAC*, 2003.

[11] R. Schenkel, T. Crecelius, M. Kacimi, S. Michel, T. Neumann, J. X. Parreira, and G. Weikum. Efficient top-k querying over social-tagging networks. In *SIGIR*, 2008.

[12] T. Schleyer, H. Spallek, B. Butler, S. Subramanian, D. Weiss, M. Poythress, P. Rattanathikun, and G. Mueller. Facebook for scientists: requirements and services for optimizing how scientific collaborations are established. *JMIR*, 10(3), 2008.

[13] Y. Sun, R. Barber, M. Gupta, C. Aggarwal, and J. Han. Co-author relationship prediction in heterogeneous bibliographic networks. In *ASONAM*, 2011.