A Unified Framework for Link Recommendation Using Random Walks

Zhijun Yin, Manish Gupta, Tim Weninger, Jiawei Han
Department of Computer Science, University of Illinois at Urbana-Champaign
{zyin3, gupta58, weninge1, hanj}@illinois.edu

Abstract—**The phenomenal success of social networking sites, such as Facebook, Twitter and LinkedIn, has revolutionized the way people communicate, and this paradigm have attracted the attention of researchers that wish to study the social and technological problems that arise. One such problem is that of link recommendation which is a critical task that not only helps improve the user experience but also is essential to network growth itself. In an effective link recommendation algorithm it is essential to identify the factors that influence link creation. This paper enumerates several of these well studied factors and then proposes an approach which satisfies these factors. This approach estimates link relevance by estimating the link relevance by using using random walk algorithm on an augmented social graph with both attribute and structure information. The global and local influence of the attributes is leveraged in the framework as well. Other than link recommendation, our framework can also rank the attributes in the social network. Experiments on DBLP and IMDB data sets demonstrate that our method outperforms state of the art methods for link recommendation.**

I. Introduction

Social networking sites such as Facebook, Twitter, and LinkedInare becoming increasingly popular. Facebookreports to have 300 million active users, 50% of whom login every day on an average. Worldwide, more than 8 billion minutes are spent on Facebook per day. Moreover, ComScore reports claim that social networking sites account for more than 20% of all U.S. online display advertisement impressions and that online networking sites have seen dramatic increase in their user bases in recent months. The users not only use the social network sites to maintain contacts with old friends, but also use the sites to find new friends with similar interests and for business networking. It is reported that an average user has 130 friends on Facebook. Since the link among people is the underlying key concept for online social network sites, it is not surprising that link recommendation is an essential link mining task. First, link recommendation can help users to find potential friends, a function that improves user experience in social networking sites and attracts more users consequently. Compared with the usual passive ways of locating possible friends, the users on these social networks are provided with a list of potential friends, with a simple confirmation click. Second, link recommendation helps the social networking sites grow fast. A more complete social graph not only improves user involvement, but also provides the monetary benefits associated with a wide user base such as a large publisher network for advertisements.

Link prediction is the problem of predicting the existence of a link between two entities in an entity relationship graph, where prediction is based on the attributes of the objects and other observed links. Link prediction has been studied on various kinds of graphs including metabolic pathways, protein-protein interaction, social networks, etc. These studies use different measures such as node-wise similarity and topology-based similarity to predict the existence of the links. In addition to these existing measures, different models have been investigated for the link prediction tasks including relational Bayesian networks and relational Markov networks. Link recommendation in social network is closely related to link prediction, but has its own specific properties. Social network can be considered as a graph where each node has its own attributes. Linked entities share certain similarities with respect to attribute information associated with entities and structure information associated with the graph. We study the problem of expressing the link relevance to incorporate both attributes and structure in a unified intuitive manner.

In this paper, we propose a framework using both attribute and structural properties to recommend potential linkages in social networks. To compute accurate link recommendations in social networks, we propose a list of desired criteria. A random walk framework on the augmented social graphs using both attribute and structural properties is further proposed, which satisfies all the criteria. We also discuss different methods for setting edge weights in the augmented social graph which considers both global and local characteristics of the attributes. Extensive experiments have been performed on two real data sets: DBLP and IMDB. We show that our method performs significantly better than state of the art methods.

The contributions of the paper are summarized as follows.

- We propose several desired criteria of link recommendation in social networks, and demonstrate these criteria in real data sets.
- A unified link recommendation framework based on both attributes and structure is proposed which satisfies the desired criteria of link recommendation.
- Several methods are used for edge weighting in the augmented social graph. Both global and local information of the attributes has been leveraged in the framework.
- Our framework can also rank the attributes personalized to a particular person node.
- Extensive experiments have been conducted on DBLP and IMDB data sets.

II. Problem Formulation

Given a *social graph* G (V,E), where V is the set of nodes and E is the set of edges, each node in V represents a person in the network and each edge in E represents a link between two person nodes. Besides the links, each person has his/her own attributes. The existence of an edge in G represents a link relationship between the two persons.

The *link recommendation task* can be expressed as: given node v in V, provide a ranked list of nodes in V as the potential links ranked by link relevance (with the existing linked nodes of v removed).

The following presents some intuition-based desiderata for link relevance where *Alice* is more likely to form a link with *Bob* rather than with *Carol*.

1) *Homophily*: Two persons who share more attributes are more likely to be linked than those who share fewer attributes. E.g., *Alice* and *Bob* both like *Football* and *Tennis*, and *Alice* has no common interest with *Carol*.
2) *Rarity*: The rare attributes are likely to be more important, whereas the common attributes are less important. E.g., only *Alice* and *Bob* love *Hiking*, but thousands of people, including *Alice* and *Carol*, are interested in *Football*.
3) *Social influence*: The attributes shared by a large percentage of friends of a particular person are important for predicting potential links for that person. E.g., most of the people linked to *Alice* like *Football*, and *Bob* is interested in *Football* but *Carol* is not.
4) *Common friendship*: The more neighbors two persons share, the more likely it is that they are linked together. E.g., *Alice* and *Bob* share over one hundred friends, but *Alice* and *Carol* have no common friend.
5) *Social closeness*: The potential friends are likely to be located close to each other in the social graph. E.g., *Alice* and *Bob* are only one step away from each other in social graph, but *Alice* and *Carol* are five steps apart.
6) *Preferential attachment*: A person is more likely to link to a popular person rather than to a person with only a few friends. E.g., *Bob* is very popular and has thousands of friends, but *Carol* has only ten friends.

A good link candidate should satisfy the above criteria both on the attribute and structure in social networks. In other words, the link relevance should be estimated by considering the above intuitive rules.

III. Proposed Solution

A. Graph Construction

Given the original social graph $G(V,E)$, we construct a new graph $G'(V',E')$, augmented based on G. Specifically, for each node in graph G, we create a corresponding node in G', called *person node*. For each edge in E in graph G, we create a corresponding edge in G'. For each attribute a, we create an additional node in G', called *attribute node*. V' consists of V_p and V_a, where V_p is the person node set and V_a is the attribute node set. For every attribute of a person, we create a corresponding edge between the person node and the attribute node.

TABLE I
The attributes and relationships of the users in the social network

User	Attributes	Friends
Alice	“c++”, “python”	Bob, Carol
Bob	“c++”, “c#”, “python”	Alice, Carol
Carol	“c++”, “c#”, “perl”	Alice, Bob, Dave
Dave	“java”, “perl”	Carol, Eve
Eve	“java”, “perl”	Dave

Example 1 Consider a social network of five people: *Alice*, *Bob*, *Carol*, *Dave* and *Eve*. The attributes and relationships of the users are shown in Table I. The augmented graph G' containing both person nodes and attribute nodes is shown in Figure 1.

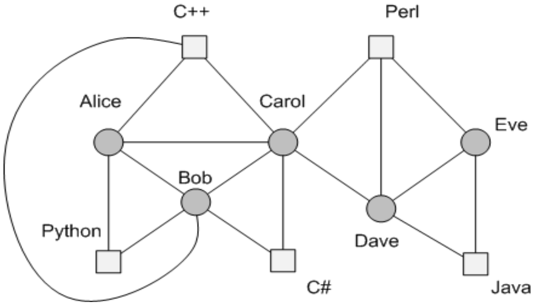


Fig. 1. The augmented graph with person and attribute nodes (round nodes are person nodes and square nodes are attribute nodes)

The edge weights in G' are defined by the uniform weighting scheme. The weight $w(a,p)$ of the edge from attribute node a to person node p is defined as follows.

$$w(a,p) = \frac{1}{|N_p(a)|} \quad (1)$$

where $N_p(a)$ denotes the set of person nodes connected to attribute node a.

Given person node p, attribute node a connected to p and person node p' connected to node p, the edge weight $w(p,a)$ from person node p to attribute node a and the edge weight $w(p,p')$ from person node p to person node p' are defined as follows.

$$w(p,a) = \begin{cases} \frac{\lambda}{|N_a(p)|} & \text{if } |N_a(p)| > 0 \text{ and } |N_p(p)| > 0; \\ \frac{1}{|N_a(p)|} & \text{if } |N_a(p)| > 0 \text{ and } |N_p(p)| = 0; \\ 0 & \text{otherwise.} \end{cases} \quad (2)$$

$$w(p,p') = \begin{cases} \frac{1-\lambda}{|N_p(p)|} & \text{if } |N_p(p)| > 0 \text{ and } |N_a(p)| > 0; \\ \frac{1}{|N_p(p)|} & \text{if } |N_p(p)| > 0 \text{ and } |N_a(p)| = 0; \\ 0 & \text{otherwise.} \end{cases} \quad (3)$$

where $N_a(p)$ denotes the set of the attribute nodes connected to node p and $N_p(p)$ denotes the set of person nodes connected to node p.

Here, λ controls the tradeoff between attribute and structural properties. The larger λ is, the more the algorithm uses attribute properties for link recommendation. Specifically, if $\lambda = 1$, the algorithm makes use of the attribute features only. If $\lambda = 0$, it is based on structural properties only.

B. Algorithm Design

In order to calculate the link relevance based on the criteria in Section II, we propose a random walk based algorithm on the newly constructed graph to simulate the friendship hunting behavior. The stationary probabilities of random walk starting from a given person node are considered as the link relevance between the person node and the respective nodes in the probability distribution.

Random walk process on the newly constructed graph satisfies the desiderata (provided in the Section II) for link relevance in the following ways.

1) *Homophily*: If two persons share more attributes, the corresponding person nodes in the graph will have more connected attribute nodes in common. Therefore, the random walk probability from one person node to the other is high.
2) *Rarity*: If one attribute is rare, there are fewer outlinks for the corresponding attribute node. The weight of each outlink is larger, because there are fewer outlinks. Therefore, the probability of a random walk originating from a person and reaching the other person node via this attribute node is larger, which implies that the rare attribute plays a more important role than the popular attribute.
3) *Social influence*: If one attribute is shared by many of the existing linked persons of the given person, the random walk will pass through the existing linked person nodes to this attribute node. Therefore, the random walk probability from the given person node to this attribute node is large, so this attribute is likely to be important to the given person for link recommendation.
4) *Common friendship*: If two persons share many friends, these two person nodes have a large number of common neighbors in the graph. Therefore, the random walk probability from one person to the other is high.
5) *Social closeness*: If two persons are close to each other in the graph, the random walk probability from one to the other is likely to be larger than if they are far away from each other.
6) *Preferential attachment*: If a person is very popular and links to many persons, there are many inlinks to the person node in the graph. For a random person node in the graph it is easier to access a node with more inlinks.

Here, we use the random walk with restart on the graph to calculate the link relevance for a particular person p^*.

$$\begin{aligned} r_p &= (1-\alpha) \sum_{p' \in N_p(p)} w(p',p) r_{p'} \\ &\quad + (1-\alpha) \sum_{a' \in N_a(p)} w(a',p) r_{a'} + \alpha r_p^{(0)} \end{aligned} \tag{4}$$

$$r_a = (1-\alpha) \sum_{p' \in N_p(p)} w(p',a) r_{p'} \tag{5}$$

where r_p is the link relevance of person p with regard to p^*, i.e., the random walk probability of person node p from person node p^*. r_a is the relevance of attribute a with regard to p^*, i.e., the random walk probability of attribute node a from person node p^*. α is the restart probability. $r_p^{(0)} = 1$ if node p refers to person p^* and $r_p^{(0)} = 0$ otherwise.

The link recommendation algorithm is listed as follows.

Algorithm 1 Link Recommendation

```
Input: A social graph $G(V,E)$, person attribute profile, person $p^*$, and two parameters $\lambda$ and $\alpha$
Output: A ranked list of recommended candidates (to be linked with) for person $p^*$
1) Construct the augmented graph $G'(V',E')$ based on social graph $G(V,E)$ and person attribute profile, where $V' = V_p \cup V_a$, $V_p$ is the person node set, $V_a$ is the attribute node set.
2) Set the edge weights with $\lambda$ in the augmented graph $G'$ using Equation 1, 2 and 3.
3) Iterate to update $r_p$ and $r_a$ according to Equation 4, 5 for person node $p \in V_p$ and attribute node $a \in V_a$ until convergence. $r_p^{(0)} = 1$ if $p = p^*$ and $r_p^{(0)} = 0$ otherwise.
4) Let $r_p^*$ be the stationary value for $r_p^{(t)}$. Output the nodes in $V_p$ based on the non-increasing order of $r_p^*$, where the nodes connected to $p^*$ in $G$ are excluded.
```

IV. Discussion

A. Edge Weighting

The edge weighting in the augmented graph is important to the link recommendation algorithm. In Section III-A, we assigned weights to each attribute equally. Here we propose several edge weighting methods for the edges from person nodes to attribute nodes. The edge weight $w(p,a)$ from person node p to attribute node a is as follows.

$$w(p,a) = \begin{cases} \frac{\lambda w_p(a)}{\sum_{a' \in N_a(p)} w_p(a')} & \text{if } |N_a(p)| > 0 \text{ and } |N_p(p)| > 0; \\ \frac{w_p(a)}{\sum_{a' \in N_a(p)} w_p(a')} & \text{if } |N_a(p)| > 0 \text{ and } |N_p(p)| = 0; \\ 0 & \text{otherwise}. \end{cases}$$

where $w_p(a)$ is the importance score for attribute a with regard to person p. $N_a(p)$ denotes the set of the attribute nodes connected to node p. $N_p(p)$ denotes the set of the person nodes connected to node p. λ controls the tradeoff between attribute and structural properties.

Global Weighting: Instead of weighing all the attributes equally, we should attach more weight to the more promising attributes. Here we give the definition of attribute global importance $g(a)$ for attribute a in social graph $G(V,E)$ as follows.

$$g(a) = \frac{\sum_{(u,v) \in E} e_{uv}^a}{\binom{n_a}{2}}$$

n_a is the number of the persons that have attribute a. $e^a_{uv} = 1$ if person u and v both have attribute a, $e^a_{uv} = 0$ otherwise. The global importance score for attribute a measures the percentage of existing links among all the possible person pairs with the attribute a. The local importance score $g(a)$ is used as $w_p(a)$.

Local Weighting: Instead of considering the attributes globally, we derive the local importance of the attributes for the specific person based on its neighborhood. The definition of attribute local importance $l_p(a)$ for attribute a with regard to person p is as follows.

$$l_p(a) = \sum_{p' \in N_p(p)} A(p', a)$$

$N_p(p)$ denotes the set of the person nodes connected to node p. $A(p, a) = 1$ if person p has attribute a, $A(p, a) = 0$ otherwise. The definition demonstrates that the more the number of friends that share the attribute, the more important the attribute is for the person. The local importance score $l_p(a)$ is used as $w_p(a)$, so the edge weight from person p to attribute a depends on the local importance of a with regard to p.

Mixed Weighting: Other than considering global and local importance separately, we can combine the two together.

The first mixture method is to use linear interpolation to combine the global and local importance together.

$$w_p(a) = \gamma \frac{g(a)}{\sum_{a' \in N_a(p)} g(a')} + (1 - \gamma) \frac{l_p(a)}{\sum_{a' \in N_a(p)} l_p(a')}$$

γ controls the tradeoff between the global importance score and the local importance score.

The second mixture method is to construct attribute importance score by multiplying global and local importance score.

$$w_p(a) = g(a) l_p(a)$$

B. Attribute Ranking

Besides link recommendation, we can rank attributes with respect to a specific person by using the proposed framework. Attribute ranking can have many potential applications. For example, advertisements can be targeted more accurately if we know a person's interests more precisely. Furthermore, we can analyze the behavior of users of a particular category.

In the augmented graph, all the nodes including the attribute nodes have the random walk probability. Similarly, we can rank attribute nodes based on the random walk probability in Equation 5. The attributes with high ranks in our framework are those that can be easily accessed by the given person, the existing friends and the potential friends.

Instead of ranking the attributes for a single person, we can also rank the attributes for a cluster of person nodes. For example, we can discover the most relevant interests for all computer science graduate students. To achieve this, instead of starting random walk from a single node, we can restart with a bundle of nodes. The equations are the same as Equation 4 and 5 except for the definition of $r_p^{(0)}$. Let P be the set of the persons to be analyzed, $r_p^{(0)} = \frac{1}{|P|}$ if node p belongs to P and $r_p^{(0)} = 0$ otherwise.

C. Complexity and Efficiency Issue

The main part of the algorithm is based on the random walk process represented by Equation 4 and 5. Every time the random walk probability is updated from the neighbor nodes, so the complexity of the algorithm is $O(iter|E'|)$ where $iter$ is the iteration numbers and $|E'|$ is the edge count of the augmented graph G'. To further improve the efficiency, we can adopt the fast random walk technique in [18]. Moreover, instead of calculating the random walk with restart probability for the given node on the whole graph, we can extract the surrounding k-hop nodes and run the algorithm on the local graph. In the experiments we also show that large α is preferred because link recommendation depends on the neighborhood information heavily. Large α leads to fast convergence speed, and the top recommended links are stable after only a few steps.

V. Experiment

In this section, we describe our experiments on real data sets to demonstrate the effectiveness of our framework.

A. Data Sets

We use two real data sets DBLP and IMDB. For DBLP[1], we use the authors in the WWW conferences from 2001 to 2008 as persons and terms in their paper titles as attributes. Each person has ∼93 attributes and ∼7 links on average. Co-authorship prediction is considered as link recommendation problem for DBLP. For IMDB[2], we take all the actors/actresses who have performed in more than 10 movies since 2005 and consider movie locations as their attributes. There are 6750 persons and 9851 attributes. Each person has ∼29 attributes and ∼97 links on average. Co-staring prediction is considered as link recommendation problem for IMDB.

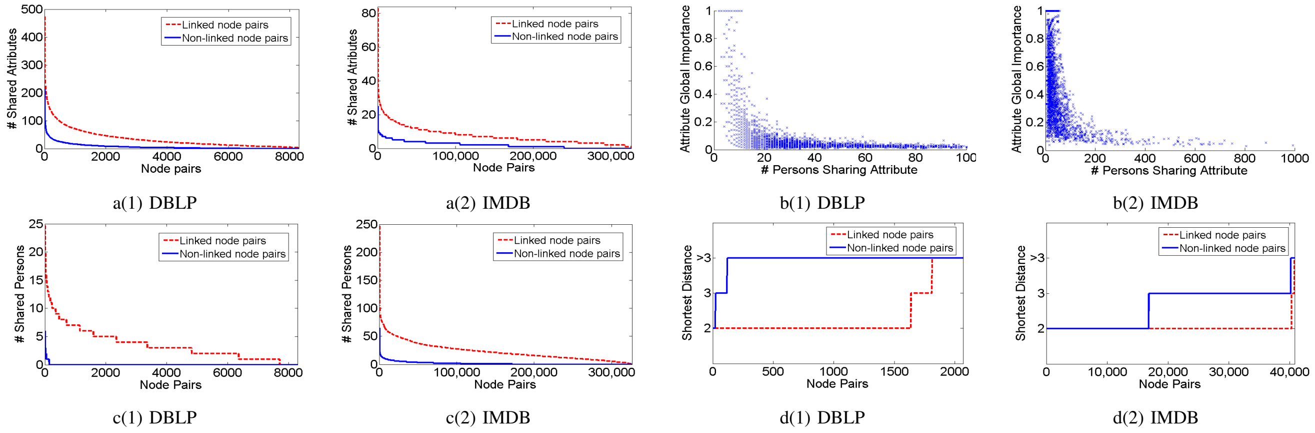
B. Link Recommendation Criteria

We proposed the desired criteria for link recommendation. Here we show the existence of these criteria in both data sets.

1) *Homophily*: We sample the same number of non-linked pairs as that of linked pairs in both data sets. As shown in Figures 2a(1) and 2a(2), compared to the non-linked pairs, the linked pairs are more likely to share more attributes.
2) *Rarity*: We analyze the correlation between the global importance of an attribute and the number of people sharing the attribute. The global importance of the attribute measures the percentage of existing links among all the possible person pairs with this attribute. The larger the global weight is, the more predictive the attribute is for link recommendation. As shown in Figures 2b(1) and 2b(2), we find that the attributes of lower frequency are likely to have higher global weights.

[1] http://www.informatik.uni-trier.de/∼ley/db/

[2] http://www.imdb.com



a(1) DBLP

a(2) IMDB

b(1) DBLP

b(2) IMDB

c(1) DBLP

c(2) IMDB

d(1) DBLP

d(2) IMDB

Fig. 2. Justification of link recommendation criteria in DBLP and IMDB

3) *Social influence*: If we randomly draw a person from the linked persons, it is obvious that the selected person is more likely to have the frequent attribute in common with these linked persons.
4) *Common friendship*: We sample equal number of non-linked pairs and linked pairs. As shown in Figures 2c(1) and 2c(2), compared to the non-linked pairs, the linked pairs are more likely to share more neighbors.
5) *Social closeness*: We construct a new graph by removing 25% linked node pairs from the original graph. We test the distances between the removed 25% node pairs in the new graph. We sample the same number of non-linked pairs as the removed linked node pairs in the original graph. As shown in Figures 2d(1) and 2d(2), compared to the non-linked pairs in the original graph, these 25% node pairs are much closer to each other.
6) *Preferential attachment*: The node degree infers how many people a particular person is linked to. A popular person is more likely to be highly linked.

C. Accuracy Metrics and Baseline

Accuracy Metrics. We remove some of the edges in the graph and recommend the links based on the pruned graph. Four-fold cross validation is used on both of the data sets in the experiment: divide the set of links in the social graph into four partitions, use one partition for testing, and retain the links in other partitions. We randomly sample 100 people and recommend the top-k links for each person. We use precision, recall and mean reciprocal rank (MRR) for reporting accuracy. P@k $= \frac{1}{|S|}\sum_{p \in S} P_k(p)$ where S is the set of sampled person nodes, $P_k(p) = \frac{N_k(p)}{k}$ and $N_k(p)$ is the number of the truly linked persons in the top-k list of person p. recall $= \frac{1}{|S|}\sum_{p \in S} recall(p)$ where $recall(p) = \frac{|F_p \cap R_p|}{|F_p|}$ (recall is measured on the top-50 results). F_p is the truly linked person set of person p and R_p is the set of recommended linked persons of person p. MRR $= \frac{1}{|S|}\sum_{p \in S} \frac{1}{rank_p}$ where $rank_p$ is the rank of the first correctly recommended link of person p.

Baseline methods To demonstrate the effectiveness of our method, we compare our method with the other methods based on the attribute and structure.

- Random: Randomly selection
- SimAttr: Cosine similarity based on the attribute space
- WeightedSimAttr: Cosine similarity based on the attribute space using global importance as the attribute weight
- ShortestDistance: The length of the shortest path.
- CommonNeighbors: $score(x, y) = |\Gamma(x) \cap \Gamma(y)|$. $\Gamma(x)$ be the set of neighbors of x in graph G.
- Jaccard: $score(x, y) = |\Gamma(x) \cap \Gamma(y)|/|\Gamma(x) \cup \Gamma(y)|$.
- Adamic/Adar: $score(x, y) = \sum_{z \in \Gamma(x) \cap \Gamma(y)} \frac{1}{\log|\Gamma(z)|}$.
- PrefAttach: $score(x, y) = |\Gamma(x)| \cdot |\Gamma(y)|$
- Katz: $score(x, y) = \sum_{l=1..\infty} \beta^l \cdot |path_{x,y}^{<l>}|$, where $path_{x,y}^{<l>}$ is the set of all length-l paths from x to y. Here we consider the paths with length no more than 3.

To compare our method with the supervised learning methods, we use Support Vector Machine (SVM) on a combination of attribute and structure features. Specifically, we use the promising features, including SimAttr, WeightedSimAttr, CommonNeighbors, Jaccard, Adamic/Adar and Katz, for the training. Here we use the LIBSVM toolkit[3]. Both linear kernel and Radial Basis Function (RBF) kernel are tested. We use SVM_Linear to denote the SVM method using linear kernel and SVM_RBF to denote the SVM method using RBF kernel in Table II and Table III.

We use RW_Uniform to denote our method using uniform weighting scheme, RW_Global to denote our method using global edge weighting, RW_Local to denote our method using local edge weighting, RW_MIX to denote our method using mixed weighting of global and local importance by linear interpolation, and RW_MIX2 to denote our method using mixed weighting by multiplication of global and local attribute importance.

D. Methods Comparison

Here we compare accuracy of link recommendation using different methods on the DBLP and IMDB data sets. The results are listed in Tables II and III. Random method performs the worst as expected. Since there are so many persons in the graph, it is almost impossible to recommend the correct

[3] http://www.csie.ntu.edu.tw/˜cjlin/libsvm

TABLE II
COMPARISON OF THE METHODS ON THE DBLP DATA SET

	P@1	P@5	P@10	P@20	P@50	Recall	MRR
Random	0.0000	0.0000	0.0010	0.0019	0.0017	0.0244	0.0042
PrefAttach	0.0225	0.0150	0.0145	0.0111	0.0090	0.1187	0.0570
ShortestDistance	0.0750	0.0655	0.0603	0.0538	0.0376	0.7050	0.1833
SimAttr	0.3625	0.1455	0.0950	0.0603	0.0325	0.5791	0.4478
WeightedSimAttr	0.6175	0.2805	0.1718	0.0974	0.0452	0.7379	0.6744
CommonNeighbors	0.5775	0.2725	0.1708	0.1028	0.0505	0.8155	0.6646
Jaccard	0.5625	0.2720	0.1708	0.1048	0.0500	0.7998	0.6540
Adamic/Adar	0.6275	0.2985	0.1873	0.1094	0.0513	0.8226	0.7127
Katz $\beta = 0.05$	0.5750	0.2650	0.1745	0.1039	0.0505	0.8196	0.6636
Katz $\beta = 0.005$	0.5725	0.2675	0.1755	0.1045	0.0505	0.8188	0.6641
Katz $\beta = 0.0005$	0.5725	0.2665	0.1755	0.1044	0.0505	0.8196	0.6630
SVM_RBF	0.5425	0.2895	0.1875	0.1096	0.0514	0.8252	0.6636
SVM_Linear	0.6225	0.2985	0.1857	0.1099	0.0511	0.8212	0.7068
RW_Uniform $\lambda = 0.6$ $\alpha = 0.9$	0.7000	0.3470	0.2137	0.1228	**0.0555**	**0.9068**	0.7767
RW_Global $\lambda = 0.6$ $\alpha = 0.7$	0.7350	0.3530	0.2175	**0.1239**	0.0553	0.8912	0.7950
RW_Local $\lambda = 0.7$ $\alpha = 0.9$	0.7225	0.3345	0.1990	0.1135	0.0520	0.8589	0.7882
RW_MIX $\lambda = 0.6$ $\alpha = 0.9$ $\gamma = 0.6$	**0.7475**	**0.3605**	**0.2187**	0.1224	0.0547	0.8809	**0.8058**
RW_MIX2 $\lambda = 0.5$ $\alpha = 0.9$	0.7200	0.3455	0.2058	0.1185	0.0539	0.8727	0.7870

TABLE III
COMPARISON OF THE METHODS ON THE IMDB DATA SET

Method	P@1	P@5	P@10	P@20	P@50	Recall	MRR
Random	0.0025	0.0035	0.0030	0.0024	0.0030	0.0076	0.0119
PrefAttach	0.0475	0.0280	0.0233	0.0216	0.0173	0.0273	0.0915
ShortestDistance	0.0025	0.0080	0.0072	0.0078	0.0088	0.0319	0.0325
SimAttr	0.6625	0.5355	0.4240	0.2914	0.1592	0.3606	0.7384
WeightedSimAttr	0.8175	0.6820	0.5648	0.4140	0.2177	0.4763	0.8524
CommonNeighbors	0.8475	0.7395	0.6525	0.5044	0.2870	0.6390	0.8999
Jaccard	0.8775	0.7705	0.6835	0.5473	0.3151	0.6694	0.9134
Adamic/Adar	0.8450	0.7570	0.6695	0.5184	0.2991	0.6655	0.8992
Katz $\beta = 0.05$	0.4200	0.3665	0.3335	0.2594	0.1552	0.3564	0.5307
Katz $\beta = 0.005$	0.7425	0.6710	0.5840	0.4446	0.2544	0.5763	0.8332
Katz $\beta = 0.0005$	0.8175	0.7160	0.6343	0.4854	0.2765	0.6064	0.8776
SVM_RBF	0.7450	0.6955	0.6342	0.5149	0.3047	0.6768	0.8226
SVM_Linear	0.8550	0.7590	0.6788	0.5531	0.3314	0.7119	0.9002
RW_Uniform $\lambda = 0.1$ $\alpha = 0.8$	0.8775	0.7660	0.6832	0.5544	0.3329	**0.7243**	0.9172
RW_Global $\lambda = 0.2$ $\alpha = 0.9$	0.9100	0.7985	0.6935	0.5508	**0.3354**	0.7234	0.9382
RW_Local $\lambda = 0.4$ $\alpha = 0.9$	0.9450	0.8140	0.7025	0.5429	0.3155	0.6938	0.9609
RW_MIX $\lambda = 0.4$ $\alpha = 0.9$ $\gamma = 0.1$	0.9425	0.8125	0.7035	0.5431	0.3183	0.6986	0.9592
RW_MIX2 $\lambda = 0.2$ $\alpha = 0.9$	**0.9525**	**0.8180**	**0.7058**	**0.5591**	0.3349	0.7230	**0.9648**

links by random selection. PrefAttach and ShortestDistance performs poorly in both data sets.

Structure-based measures other than ShortestDistance perform well for both data sets. This indicates that the graph structure plays a crucial role in link recommendation. Compared with DBLP, precision and MRR in IMDB are much higher, but recall is lower. The reason is that on average there are much more links per person in IMDB (96.67) than in DBLP (6.63). The more the links, the more likely we can get correct link recommendations in the top results. Furthermore, dense graph structure makes structure-based measures more expressive.

Attribute-based measures (especially WeightedSimAttr) perform fairly well in both DBLP and IMDB. Accuracy achieved by WeightedSimAttr is comparable to that achieved by structure-based measures. It indicates that attribute information compliments to the structure features for link recommendation in these two data sets. WeightedSimAttr uses the global importance as the attribute weight, whereas SimAttr weighs all the attributes equally. The effectiveness of global importance score helps WeightedSimAttr to be more accurate than SimAttr.

Supervised learning methods SVM_RBF and SVM_Linear perform well, but cannot beat the best baseline measure in precision at top in both of the data sets. It shows that directly combining attribute and structure features using supervised learning technique may not lead to good results. Although SVM makes use of both attribute and structure properties, it does not take into account the semantics behind the link recommendation criteria when computing the model.

Compared with the baseline methods, our methods perform significantly better in both DBLP and IMDB. In DBLP, RW_MIX has the best precision (74.75% precision at 1, 36.05% precision at 5 and 21.87% precision at 10) and the best MRR (80.58%), while RW_Uniform has the best recall (90.68%). In IMDB, RW_MIX2 has the best precision (95.25% precision at 1, 81.80% precision at 5, 70.58% precision at 10) and the best MRR (96.48%), while RW_Uniform has the best recall (72.43%). Global and local weighting methods reinforce the link recommendation criteria. Hence, RW_Global, RW_Local, RW_MIX and RW_MIX2 can beat RW_Uniform in terms of precision at top and MRR. In DBLP, RW_Global performs better than RW_Local, because the global attributes (keywords) play an important role in link recommendation compared to very specific attributes shared with coauthors. In IMDB, RW_Local performs better than RW_Global, which suggests that the movie locations of the partners has a significant influence on actors. Also, in DBLP, RW_MIX can beat both RW_Global and RW_Local, whereas in IMDB RW_MIX2 can outperform RW_Global and RW_Local.

E. Parameter Setting

In our link recommendation framework, there are two parameters λ and α. We discuss how to set these two parameters and how the parameter settings affect the link recommendation results.

Parameter setting. Different data sets may lead to different optimal λ and α. We obtain the best values of these parameters by performing a grid search over ranges of values for these parameters and measuring accuracy on the validation set for each of these configuration settings.

Effect of λ setting. λ controls the tradeoff between attribute and structural properties. Higher value of λ implies that the algorithm gives more importance to the attribute features than structure features. We find the optimal λ is 0.6 in DBLP and 0.2 in IMDB, and the combination of attribute and structural features is much better than using attribute or structure properties individually.

Effect of α setting. α is the restart probability of random walks. Random walk with restart is quite popular in applications like personalized search and query suggestion. In our link recommendation setting, large α provides more accurate link recommendation, unlike low α in traditional applications. In personalized search, random walks are used to discover relevant entities spread out in the entire graph, so a small α is favorable in these cases. However, in link recommendation task, we are more focused on the local neighborhood information, so a large α is more reasonable. We find that α = 0.9 provides the best result. Besides high accuracy, large α makes the algorithm converge fast.

F. Case study

We select several well known researchers and show the recommended persons in Table IV as well as top ranked keywords for each person in Table V. Since we partition the links into four partitions, the recommended persons in Table IV are selected from top-3 results in each partition obtained by applying our framework using global weighting strategy. The top ranked keywords in Table V are selected by applying our framework using uniform weighting on the complete coauthorship graph without partitioning.

VI. Related work

In [4], [5], Getoor et al. classify link mining tasks into three types: object-related, link-related, and graph-related. Node-wise similarity based methods try to seek an appropriate distance measure for two objects. In [2], the authors estimate the weight values from a set of linear regression equations obtained from a social network graph that captures human judgment about similarity of items. We define a more intuitive way of defining the link prediction ability of each attribute. In [8], the authors use node information for link and link-type prediction on metabolic pathway, protein-protein interaction and coauthorship datasets. They use label propagation over pairs of nodes with multiple link types and predict relationships among the nodes. We use random walks for link recommendation.

Topological pattern based methods focus on exploiting either local or global patterns that could well describe the network. In [1], Chen et al. present a data clustering algorithm K-destinations using random walk hitting time on directed graphs. Nodes within a cluster can be considered as friends. In [3], the authors use random-walk related quantities like square root of the average commute time and the psuedo-inverse of the Laplacian matrix to compute similarity between nodes. In [10], Nowell and Kleinberg suggest that link predictions could be done using network topology alone. They present results on five coauthorship networks using features like common neighbors, Jaccard's coefficient, Adamic/Adar, preferential attachment, hitting time, commute time, and simrank. In [11], they also suggest using meta-approaches like low rank approximation, unseen bi-grams and clustering besides the above features. We use a random walk model and accuracy measures as used in [10].

Probabilistic inference helps capture the correlations among the links. But exact inferences are intractable and so approximate inference is done. These models often need domain knowledge and are difficult to interpret. High computational cost restricts their applicability to contemporary large-sized networks. Probabilistic model based approaches have been discussed in [7], [9], [15] and [16].

Some works have combined the above mentioned approaches. In [6], Hasan et al. identify mix of node and graph structure features for supervised learning using SVMs, decision trees and multilayer perceptrons to predict coauthorship relationships. In [12], the authors learn classifiers like logistic regression and naive bayes for predicting temporal link using both network and the entity features. In [13], Popescul and Ungar propose the usage of statistical relational learning to build link prediction models. In [14], Rattigan and Jensen demonstrate the effectiveness of link prediction models to solve the problem of anomalous link discovery. Zhou et al. in [17] propose a graph clustering algorithm (similar to k-medoids) based on both structural and attribute similarities through a unified distance measure. They then learn the degree of contributions of structural similarity and attribute similarity. For our link recommendation task, it is crucial to

TABLE IV
RECOMMENDED PERSONS IN DBLP (THE TRULY LINKED PERSONS ARE IN *Italics*)

Rakesh Agrawal	Ricardo A. Baeza-Yates	Jon M. Kleinberg	Ravi Kumar	Gerhard Weikum
Roberto J. Bayardo Jr.	*Nivio Ziviani*	*Christos Faloutsos*	*Andrew Tomkins*	*Fabian M. Suchanek*
Ramakrishnan Srikant	*Carlos Castillo*	*Jure Leskovec*	*D. Sivakumar*	*Gjergji Kasneci*
Jerry Kiernan	*Vassilis Plachouras*	*Prabhakar Raghavan*	*Andrei Z. Broder*	*Klaus Berberich*
Christos Faloutsos	*Álvaro R. Pereira Jr.*	*Andrew Tomkins*	*Sridhar Rajagopalan*	*Srikanta J. Bedathur*
Yirong Xu	*Massimiliano Ciaramita*	*Ravi Kumar*	*Ziv Bar-Yossef*	*Michalis Vazirgiannis*
Daniel Gruhl	*Aristides Gionis*	*Cynthia Dwork*	*Prabhakar Raghavan*	*Stefano Ceri*
Gerhard Weikum	*Barbara Poblete*	*Lars Backstrom*	*Jasmine Novak*	*Timos K. Sellis*
Timos K. Sellis	*Gleb Skobeltsyn*	*Ronald Fagin*	*Jon M. Kleinberg*	*Jennifer Widom*
Serge Abiteboul	Ravi Kumar	*Sridhar Rajagopalan*	*Christopher Olston*	*Hector Garcia-Molina*
Sridhar Rajagopalan	Massimo Santini	*Deepayan Chakrabarti*	*Anirban Dasgupta*	François Bry
Rafael González-Cabero	Sebastiano Vigna	Uriel Feige	*Daniel Gruhl*	Frank Leymann
Asunción Gómez-Pérez	Qiang Yang	D. Sivakumar	Uriel Feige	Wolfgang Nejdl

TABLE V
ATTRIBUTE RANKING IN DBLP

Rakesh Agrawal	Soumen Chakrabarti	Ricardo A. Baeza-Yates	Ravi Kumar	Jon M. Kleinberg	ChengXiang Zhai	Jure Leskovec	Gerhard Weikum
mining	search	search	search	networks	retrieval	networks	xml
database	mining	retrieval	networks	algorithms	information	graphs	search
databases	information	information	information	search	search	information	information
information	algorithms	query	algorithms	social	language	graph	database
systems	dynamic	semantic	time	information	mining	network	peer
search	learning	xml	semantic	network	models	social	management
applications	databases	analysis	analysis	analysis	model	evolution	query
xml	structure	model	graph	systems	learning	learning	systems
semantic	queries	searching	systems	problem	analysis	search	semantic
system	networks	matching	efficient	graph	modeling	marketing	efficient

set appropriate edge weights to satisfy the criteria mentioned in Section II. We learn personalized edge weights for different attributes in a local versus global setting.

VII. CONCLUSIONS AND FUTURE WORK

In this paper, we proposed a framework for link recommendation based on attribute and structural properties in a social network. We first presented some desired criteria for link recommendation. To calculate the link relevance that satisfies those criteria, we augmented the social graph with attributes as additional nodes and used the random walk algorithm on this augmented graph. Both global and local attribute information can be leveraged into the framework by influencing edge weights. Besides link recommendation, our framework can be easily adapted to provide attribute ranking as well.

Our framework can be further improved in several aspects. First, attributes may be correlated with each other. The framework should automatically identify such semantic correlations and handle it properly for link recommendation. Second, the algorithm currently adds a new attribute node for every value of categorical attributes. Handling numeric attributes would require tuning to appropriate level of discretization. We also plan to test the effectiveness of our method on friendship networks like Facebook.

REFERENCES

[1] M. Chen, J. Liu, and X. Tang. Clustering via random walk hitting time on directed graphs. In *AAAI*, pages 616–621, 2008.

[2] S. Debnath, N. Ganguly, and P. Mitra. Feature weighting in content based recommendation system using social network analysis. In *WWW*, pages 1041–1042, 2008.

[3] F. Fouss, A. Pirotte, J.-M. Renders, and M. Saerens. Random-walk computation of similarities between nodes of a graph with application to collaborative recommendation. *IEEE Trans. Knowl. Data Eng.*, 19(3):355–369, 2007.

[4] L. Getoor. Link mining: a new data mining challenge. *SIGKDD Explorations*, 5(1):84–89, 2003.

[5] L. Getoor and C. P. Diehl. Link mining: a survey. *SIGKDD Explorations*, 7(2):3–12, 2005.

[6] M. A. Hasan, V. Chaoji, S. Salem, and M. Zaki. Link prediction using supervised learning. In *SDM 06 workshop on Link Analysis, Counterterrorism and Security*, 2006.

[7] H. Kashima and N. Abe. A parameterized probabilistic model of network evolution for supervised link prediction. In *ICDM*, pages 340–349, 2006.

[8] H. Kashima, T. Kato, Y. Yamanishi, M. Sugiyama, and K. Tsuda. Link propagation: A fast semi-supervised learning algorithm for link prediction. In *SDM*, pages 1099–1110, 2009.

[9] J. Kunegis and A. Lommatzsch. Learning spectral graph transformations for link prediction. In *ICML*, page 71, 2009.

[10] D. Liben-Nowell and J. M. Kleinberg. The link prediction problem for social networks. In *CIKM*, pages 556–559, 2003.

[11] D. Liben-Nowell and J. M. Kleinberg. The link-prediction problem for social networks. *JASIST*, 58(7):1019–1031, 2007.

[12] J. O'Madadhain, J. Hutchins, and P. Smyth. Prediction and ranking algorithms for event-based network data. *SIGKDD Explorations*, 7(2):23–30, 2005.

[13] A. Popescul, R. Popescul, and L. H. Ungar. Statistical relational learning for link prediction, 2003.

[14] M. J. Rattigan and D. Jensen. The case for anomalous link discovery. *SIGKDD Explorations*, 7(2):41–47, 2005.

[15] R. Sarukkai. Link prediction and path analysis using markov chains. *Computer Networks*, 33(1-6):377–386, 2000.

[16] B. Taskar, M. F. Wong, P. Abbeel, and D. Koller. Link prediction in relational data. In *NIPS*, 2003.

[17] Y. Zhou, H. Cheng, and J. X. Yu. Graph clustering based on structural/attribute similarities. *PVLDB*, 2(1):718–729, 2009.

[18] H. Tong, C. Faloutsos, and J.-Y. Pan. Fast random walk with restart and its applications. In *ICDM*, pp. 613–622, 2006.