Link Prediction in Social networks using Affiliation networks

Nagarajan Natarajan, vishvAs vAsuki

Contents

Abstract

Social network analysis, in particular the link prediction problem, has attracted much attention recently. In this project, we are investigating how affiliation networks and clusterings of nodes in the affiliation networks can be used in link prediction on a social network. We report upon our progress in this regard. We detail efforts at preparing suitable data for our experiments. We confirm that there is indeed some useful information in affiliation networks and affiliation network clusters, which can possibly be exploited in link prediction on the social network. But, we have not been able to outperform link prediction based on the truncated Katz similarity measure calculated from the social network alone yet. However, we propose a couple of approaches for accomplishing this.

Chapter 1

Introduction

1.1 Social and affiliation networks

A social network is a directed or undirected graph of nodes that represent people (sometimes referred to as users) and the links that represent some relationship among users.

An affiliation network is an undirected bipartite graph in which the nodes can be partitioned into two disjoint sets of entities: Users and Groups, and every edge is a link between a user and a group. An link in the affiliation network is indicative of the membership of the user in the group.

1.1.1 Terminology

Associated with social and affiliation networks are the adjacency matrices referred to as UserUser and UserGroup respectively. A network among users can be derived from the affiliation network by adding edges between users whenever they share an affiliation, and then removing the nodes corresponding to groups. This process is called 'folding' [2]. The folding operation on a graph G is denoted by $fold(G)$. One may similarly fold an affiliation network to derive a network among groups.

1.2 Link prediction problem

The problem of link prediction is fundamental in analysis of social networks. It has attracted serious attention in the last decade. The problem is to infer new relationships among users that are likely to develop in the future, given the current state of the network. The problem can be interpreted as inferring missing links from a given snapshot of the social network. Most of the efforts at solving this problem have relied on the social network. We try to use affiliation networks to enhance the quality of link prediction.

1.2.1 Common link prediction techniques

Link prediction methods generally assign a proximity score to each of the possible links and produce a list of scores in the descending order. *katz* is a subtle measure of proximity that out-performs many direct measures. It is explained in a later chapter. A more direct method is to assign the number of shared neighbors as the score (*Common neighbors*).

A straight-forward way to evaluate any link prediction technique would be to compare its performace against random prediction[4]. We determine the extent to which our method exploits the affiliation network by comparing against a random predictor. We also present some methods to incorporate the information from the affiliation network in a *katz*-like proximity measure, and we evaluate their performance against *katz*. We use *katz* measure as the baseline for our preliminary experiments.

1.2.2 How good is your prediction?

Two commonly used measures of quality of solutions in Information Retrieval and Classification tasks are *Precision* and *Recall*. *Precision* measures the exactness or fidelity of the solution while *Recall* measures the completeness of the solution. We define *Precision* as the ratio of the number of correct predictions to the total number of predictions made, and *Completeness* (or *Recall*) as the ratio of the number of correct predictions to the total number of possible correct predictions i.e. number of edges in the test set.

1.3 Related work

There has been prolific work on analysis of social networks in the recent past, with particular attention to link prediction. But the same cannot be said about using different sources of information for link prediction or seeking beyond the social network. Most of the existing methods for link prediction try to exploit the structure and topology of the social network. Liben-Nowell and Kleinberg[4] evaluate various measures of proximity between nodes in a social network, by predicting the links that are likely to be formed. We take a similar approach to link prediction, but we digress from their approach in the fact that we derive proximity from affiliation network. Savas et al propose a model[6] that uses multiple retroactive steps and auxiliary sources to improve link prediction. Similar to this approach, we intend to predict links in the social network using a hybrid measure of proximity: proximity in social network combined with the proximity in $fold(UserGroup)$.

1.4 Overview of the report

We begin with a chapter on our dataset. It focuses on the subtle issues in preparation of the data, obtaining the small data samples, challenges faced in

the process and the resolutions that followed. The next chapter elaborates on the preliminary experiments and results. We explain the baseline prediction methods against which we compare the performance of our methods. We describe experiments, where, using information from the affiliation network and its clusters, we show improvement in performance over the random predictor. Thereby we show that there is useful information in the affiliation network which can potentially be used in link prediction. The report concludes with an examination of the questions that remain, and with a plan of tasks we intend to complete in the coming weeks.

Chapter 2

Data

2.1 Data preparation

We expended much effort in acquiring and preparing data required for this project. For the purpose of this project, we need not only a social network among users, but also an associated affiliation network showing the memberships of users in various groups. In this section, we describe our data preparation efforts.

2.1.1 The Orkut online social network

Orkut is Google's popular online social network service. In Orkut, every user is associated with a list of friends and a list of 'communities'. This is precisely the kind of data we are looking for.

However, companies which operate such websites are reluctant to give away this data. We were unable to acquire data for our experiments directly from Google. Instead, by crawling the Orkut network and harvesting these associations, one can construct a social and an affiliation network. Mislove and coauthors did exactly this for their work [5]. They then made this data available to us at our request.

2.1.2 Reformatting the data

The main part of the data we received from Dr Mislove consisted of two text files. One text file contained a list of UserUser links, and the other contained a list of UserGroup links. There were over 3 million users and 8 million groups. We refer to this as the 'raw data'.

So, if a user with id 25 is connected to a user with id 45, in the first file, there would either be a row which read '45 25' or a row which read '25 45' or both. If a user with id 25 is connected to groups with id 400 and 500, there would be two rows in the second file which read '25 400' and '25 500'.

A graph is conveniently represented by an adjacency matrix. Furthermore, common link prediction and clustering algorithms are naturally specified as linear algebra manipulations. Furthermore, we intended to use Matlab for this project. So, our first step was to convert the raw data to two adjacency matrices corresponding to the social network and the affiliation network.

2.1.3 Cleaning up the data

As we ran our experiments, we found many inconsistencies in our assumptions about the raw data. Below, we list the data-preparation problems we encountered and solved before we could proceed with our experiments.

Even though all links in the Orkut social network are undirected, in many cases, only one directed link was recorded in the raw data, but in many other cases two directed links would be recorded. For example, if a user with id 25 is connected to a user with id 45, in the first file, there would either be a row which read '45 25' or a row which read '25 45' or both. So, we had to deal with this inconsistency in representation while preparing our adjacency matrices.

There were many users without affiliations, groups without members and users without links to other users in the social network. Link prediction and the use of affiliation network data in link prediction is the central theme of our project. So, we decided to remove such users and groups.

Maintaining data consistency

While cleaning up the data, or in extracting smaller test-networks to experiment on, we needed to remove users and groups from the network. However, in doing so, we needed to be careful in order to maintain the consistency of the network. For example, every time we remove a user from the network, it may happen that some group may become empty. Or, when we remove a group, it may turn out that some user ends up with no affiliations.

Thus, we carefully maintained the invariant that every user is linked to some other user within the network, that every group has a certain minimum number of members, and that every user is affiliated with some group. A group with no members or with just one member is not likely to yield much information about links in the social network. So, in general, we ensure that groups have at least two members.

2.1.4 Extracting small networks

Many of the clustering and link prediction algorithms we use in our experiments do not scale well to large networks. If n is the number of nodes in a network, they often have a time complexity of $O(n^2)$. As our work is exploratory in nature, we defer the problem of how to perform well on link prediction and affiliation network clustering while scaling well with the number of nodes to future work.

Extracting a small social and affiliation network pair from a large social and affiliation network pair can be done in many ways. But, one must be careful to

Dataset name	Number of users	Number of groups
socialNet10133	10133	75551

Table 2.1: A small dataset

ensure that the smaller networks preserve as faithfully as possible the important properties of the original large networks. For example, one must ensure that the small social network shows the characteristics of a social network. That is, one must be able to observe, for example, a power-law distribution of node degrees.

For this reason, naive methods such as picking the subgraph induced by a randomly and independently selected set of users does not work. This does not preserve or consider the important local structure in a social network, and as a result, the resultant graph is very sparse. One can instead cluster the large social network, and pick the small cluster. However, we choose an alternative approach. Starting at a random node in the large social network, we traverse the graph along its links, until we have visited a certain number of unique nodes. We then take the subgraph induced by this set of nodes as our small test network.

This approach for extracting a small network, in greater detail, is as follows. The input to our algorithm is the target number of nodes in the extracted graph. A node is said to be visited if it has been encountered by our algorithm. A node is said to be expanded if all its children are visited. At every step, maintain a set *visitedNodes* of nodes visited so far, and a set *expandedNodes* of expanded nodes. Initially, *visitedNodes* has only one randomly selected node, and *expandedNodes* is empty. At each step, an arbitrary (not necessarily random) node is selected from the set *visitedNodes* - *expandedNodes*, and it is expanded. All its children are added to *visitedNodes*. This process is continued until the size of *visitedNodes* exceeds the target number of nodes in the extracted graph.

We also experimented with various ways to limit the number of groups corresponding to the users in the small test network. For example, we considered the possibility of eliminating groups with small membership. However, ultimately, we found that the number of groups associated with a small set of users is not too large; so we did not find any need to further limit the number of groups associated with the small user set.

Another issue to consider while extracting a small test network from a large network is to ensure that certain good properties are satisfied. For example, for the purpose of our experiments, we don't want users without any group affiliations. Our approach to dealing with this has already been discussed in a previous section.

Given an isolated node, it is hard to predict where the next link will be formed. However, if the node is not isolated, it is possible to use the knowledge of prior links to more accurately predict where the next link will be formed. Anticipating that the use of our link prediction algorithms will be limited to such cases, we eliminated users with fewer than 2 friendship links.

We present a list of such small data-sets we use in our experiments in Table 2.1. Some properties of this and other networks are discussed in the next section.

2.2 The structure in social and affiliation networks

Understanding the properties of a social network is important in motivating and designing various link prediction and clustering methods. Ensuring that properties expected of a social network hold in a small test network is also a good 'sanity check' for the process which generated this test network.

Properties of a social network have been widely studied empirically. Based on these, many theoretical models of network formation have been proposed. Leskovec's thesis [3] is a good reference in this regard. Here we comment on some interesting properties of our social and affiliation networks.

A-priori, one might anticipate that the total number group-affiliations of users will be much smaller than the total number of users. On the contrary, we find that over 8 million groups are associated with just 3 million users.

Below we list some statistics of the socialNet10133 network.

- Group-membership count of users. Mean = 50.4, Min = 0, Mode = 6.
- Friendship link count of users. Mean = 45.4, Min = 2, Mode = 2.
- Membership count of groups. Mean = 6.7, Max = 1119, Min = 2.

Chapter 3

Preliminary Experiments and Results

3.1 The training and test sets

We use a sub-network extracted from the social network as training data. Our training data consists of 70% of edges from the social network that are randomly chosen. We use the remaining 30% edges as test data.

3.2 Baseline performance measures

3.2.1 Random predictor

Let U, E denote the sets of users and edges in the social network respectively. Let T denote the set of edges in the training set and $\bar{T}$ (i.e. $E - T$) denote the set of edges in the test set. Let p be the number of predictions made by the random predictor. The expected number of correct predictions made by the random predictor is given by $\frac{|\bar{T}|*p}{\binom{|U|}{2}-|T|}$.

3.2.2 Katz

We implemented link prediction using *katz* similarity measure on the social network. *katz* is a measure of proximity between two nodes that directly sums over the collection of paths of all lengths between the two nodes, exponentially dampened by the length. If A is the UserUser matrix and β is the *damping factor*, the matrix of *katz* scores for all pairs of nodes in the network is given by,

$$K = \sum_{i=0}^{\infty} \beta^i A^i$$

Method	*Precision*	*Completeness*
Katz	0.1873	0.1873
Random	0.0013	0.0013

Table 3.1: Baseline evaluation.

In our experiments, we computed an approximate value of *katz* by truncating the series. When the number of predicted links was equal to the number of test edges, *Completeness* measure was low. However, we found, not surprisingly, that *Completeness* increases with the increase in the ratio of number of predicted links to the number of test edges.

We can observe the values of *Precision* and *Completeness* measures for *katz* and Random predictor in Table 3.1. We also see that *katz* significantly outperforms random predictor. We see that the two measures are equal in both cases, as exactly $|\bar{T}|$ number of predictions were made. [1]

3.3 Is there information in the affiliation network for link prediction?

We now report two experiments which indicate that the affiliation network contains information useful in link prediction on the social network.

3.3.1 Using the affiliation network directly

Using the *katz* measure on $fold(UserGroup)$ performs significantly better than random prediction. We see that the prediction is 12 times more precise than random (See Table 3.4).

3.3.2 Clusters in the affiliation network

In this experiment, we address the question as to whether there is information obtained from clusters in the affiliation network. One way of clustering the affiliation network is to derive $fold(UserGroup)$ and then cluster the resulting network.

The relevance of the affiliation network clusters with respect to the social network can be analyzed. Let $\{C_i\}, i = 1, ..., K$ denote the set of K clusters identified by some clustering algorithm on $fold(UserGroup)$. We then define,

$$edgeCount(C_i, C_j) \quad = \quad \sum_{x \in C_i} \sum_{y \in C_j} UserUser_{x,y}$$

[1] This is the case for all our experiments, and hence we show only *Precision* in the rest of the report.

Network clustered	Cluster	Fraction of intra-cluster edges in Social Network
$fold(UserGroup)$	1	.988
	2	.99
UserGroup	1	.57
	2	.83

Table 3.2: What can affiliation network clusters tell us about the social network?

Number of clusters	Overall fraction of intra-cluster edges in Social Network
2	.98
3	.82
10	.42

Table 3.3: How many clusters to identify in the affiliation network for link prediction on the social network? As the number of clusters increases, the fraction of cross-cluster edges in the social network decreases.

Note that, even though the clusters are derived from the affiliation network, the edgeCount matrix is calculated using $UserUser$, not $fold(UserGroup)$.

From the edgeCount matrix, we can infer the fraction of edges in UserUser which have one end-point in C_i, and another end-point in C_j. It is natural to suppose that edges to be predicted follow the same pattern. So, we call this fraction $edgeProbability(i, j)$.

We present the $edgeProbability(i, i)$ for the clusters identified by *graclus* [1] on $fold(UserGroup)$ in Table 3.2.

As we increased the number of clusters, the $edgeProbability(i, i)$ of the identified clusters decreased significantly. This is shown in the Table 3.3.

We now propose a way of improving over random prediction by exploiting $edgeProbability$. Having clustered the nodes in $fold(UserGroup)$, and having calculated the edgeProbability matrix the method picks an edge spanning the clusters i and j with probability $edgeProbability(i, j)$.

As in the case of random prediction, we theoretically calculated the expected number of correct predictions. We see that our method performs significantly better than random prediction. We summarize our results in Table 3.4.

Experiment	Increase in *Precision* over Random prediction
katz on $fold(UserGroup)$	12 x
Random prediction biased by $UserGroup$ clustering	2.43 x

Table 3.4: Relevance of affiliation network in link prediction.

3.4 Attempts at surpassing Katz

Having confirmed that there is information in affiliation networks and in affiliation network clusters, which may be useful in link prediction on social networks, we attempted to beat *katz*, our baseline for performance in link prediction. However, our simplistic attempts haven't succeeded. So, it appears that more sophisticated techniques are needed to incorporate affiliation network information in link prediction on the social network. We propose a few such methods later in this report.

3.4.1 Using the affiliation network directly

One simple way of incorporating information from the affiliation network in link prediction could be the following:

1. $K_1 \leftarrow katz(UserUser)$.
2. $K_2 \leftarrow katz(fold(UserGroup))$.
3. Predict links with the score matrix $K_1 + K_2$.

We found that this method did not yield any improvement in *Precision* and *Completeness* compared to the predictions made with just K_1 (See Table 3.5).

3.4.2 Using affiliation network clusters

Boosting intra-cluster Katz scores

For a given clustering of nodes in the affiliation network, we introduced the notion of *edgeCount* earlier. The *predictivePower* of a cluster of nodes is defined as follows.

$$predictivePower(C_i) = \frac{edgeCount(C_i, C_i)}{\sum_{j:j \neq i} edgeCount(C_i, C_j)}$$

We tried to use *predictivePower* of the clusters in order to incorporate node clustering information in calculating a *katz*-based proximity measure among nodes. We tried to boost the proximity measure of node pairs (i, j) belonging to some common cluster C_k as follows. Having calculated the *katz* proximity measure $k(i, j)$ for all node pairs, we computed a new proximity measure $k'(i, j)$ for within-cluster node pairs: $k'(i, j) = k(i, j) * predictivePower(C_k)$.

The results

We used graclus to cluster nodes in $fold(UserGroup)$. Despite high values of *predictivePower* of the clusters as observed from Table 3.2, we found the performance of this method dropped but by a small margin compared to the performance of the baseline *katz* (See Table 3.5).

Method	*Change in Precision*
Addition of *katz* Scores	1 x
katz scores boosted using affiliation network clusters	0.99 x
Coclustering the affiliation network	0.06 x

Table 3.5: Comparison of performance of our suggested methods vs *katz*.

In another experiment, we clustered the actual affiliation network, without folding it. The clustering it yielded turned out to be less favorable for link prediction on the social network than the clustering on $fold(UserGroup)$. This is expected as the fraction of intra-cluster edges in the social network were very low (See Table 3.2). In fact, the performance was much worse compared to *katz* as seen from Table 3.5.

Chapter 4

Planned investigations

Having prepared a good data-set to test our clustering and link prediction methods on, we have so far established the following:

- There is information in the affiliation network which can be used for link prediction in the social network.
- There is some information in affiliation network clustering, which can be used in link prediction in the social network.
- The few naive attempts we tried at using information from the affiliation network, and affiliation network clustering, to improve upon the Katz measure do not work.

However, the following question remains: As far as link prediction is concerned, is there useful information in the affiliation network beyond what may be determined from the social network itself? If so, how to exploit this information to achieve performance beyond methods which use the social network alone? We will make some attempts (both simplistic and sophisticated) to settle these questions in the coming few weeks.

Savas et al[6] have proposed that using a supervised variant of the truncated Katz measure outperforms link prediction with the truncated Katz measure alone. Furthermore, they have extended this method to incorporate information from multiple sources. This method finds natural application in our case, as we are trying to incorporate information from two networks: the social and the affiliation network. So, we will see if, using a similar approach, we can outperform Katz.

Another set of questions, which remain, are the following. How can we use information from clustering in the affiliation network to out-perform link prediction using truncated Katz measure on the social network alone? If we find a good way to do this, we can then confidently compare various affiliation network co-clustering algorithms. However, ideas about how we go about answering these questions are not yet concrete.

We have reported some simplistic attempts to use information from the affiliation network clustering in boosting the performance of link prediction using the truncated Katz similarity measure. In the same spirit, we propose to try the following. We will transform the social network by assigning higher weights to intra-cluster edges, relative to cross-cluster edges, and then computing the truncated Katz similarity measure on this weighted graph, we will make our prediction. We will see if using this modified graph in calculating the similarity measure yields better performance in link prediction.

Chapter 5

Conclusion

We are investigating how affiliation networks, and clusterings of nodes in the affiliation networks can be used in link prediction on a social network. We have reported upon our progress in this regard. We detailed our efforts at preparing suitable data for our experiments. We confirmed that there is indeed some useful information in affiliation networks, which can possibly be exploited in link prediction on the social network. But, we have not been able to outperform link prediction based on the truncated Katz similarity measure calculated from the social network alone yet. However, we have proposed a couple of approaches for accomplishing this.

5.1 Acknowledgements

We thank Dr. Alan Mislove [5] for giving us the Orkut data. We referred to Berkant's and Wei's implementations of Katz and other methods. We also thank Prateek Jain and Zhengdong Lu for their insights. Finally, we thank Inderjit Dhillon for his guidance.

Bibliography

[1] Inderjit S. Dhillon, Yuqiang Guan, and Brian Kulis. Weighted graph cuts without eigenvectors a multilevel approach. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(11):1944–1957, November 2007.

[2] Silvio Lattanzi and D. Sivakumar. Affiliation networks. 2009.

[3] Jure Leskovec. *Dynamics of large networks*. PhD thesis, CMU.

[4] David Liben-Nowell and Jon Kleinberg. The link prediction problem for social networks. In *CIKM '03: Proceedings of the twelfth international conference on Information and knowledge management*.

[5] Alan Mislove, Massimiliano Marcon, Krishna P. Gummadi, Peter Druschel, and Bobby Bhattacharjee. Measurement and analysis of online social networks. October 2007.

[6] Berkant Savas, Zhengdong Lu, Wei Tang, and Inderjit S. Dhillon. Link prediction for social networks: A supervised approach. October 2009.