Using Community Information to Improve the Precision of Link Prediction Methods*

Sucheta Soundarajan† sucheta@cs.cornell.edu

John E. Hopcroft‡ jeh@cs.cornell.edu

Abstract

A great deal of work focuses on analyzing various features of networks; however, network data is often limited and incomplete. Because of this, researchers consider the link prediction problem, which asks the question of which non-existent edges in an incomplete network are most likely to actually exist in the complete network. Classical approaches to this problem compute the 'similarity' of two unconnected nodes, and conclude that highly similar nodes are the most likely to be connected in the complete network [15]. In this paper, we consider several such similarity-based measures, but supplement the similarity calculations with community information obtained from community detection algorithms. We consider a variety of datasets from multiple domains and several different methods of including community information, and show that, in most of these cases, the inclusion of community information improves the accuracy of similarity-based link prediction methods.

link prediction, social networks, communities

1 Introduction

In recent years, network analysis has become an increasingly popular topic for computer science researchers. However, much of available network data is incomplete. For example, in a network representing interactions between genes in some species, links are typically determined experimentally, and so the known links may represent fewer than 1% of the actual links [21]. Because of this, researchers sometimes wish to know which pairs of nodes that are not connected in the known network are likely to be connected in the actual network. Such knowledge is useful in cases like a gene interaction network, because it can suggest which experiments a biologist should perform to identify existing interactions. It is also useful for researchers designing or applying network algorithms, as such algorithms may perform more accurately when more links are known.

Algorithms for the link prediction problem typically compute the 'similarity' between two nodes, with the assumption that nodes that are highly similar are more likely to be connected than those that are dissimilar [15]. The research question, then, lies in the question of how best to define 'similarity.' Simple measures consider easy-to-compute factors like the number of neighbors shared between two nodes, whereas more complex definitions partition the network into groups, and then determine the probability that two nodes are connected based on the group memberships of those nodes [4, 22]. Methods of the latter type can be more accurate than those of the former type; however, they typically run very slowly and may only be practical for networks of a few thousand nodes [15].

In this paper, we consider several simple methods that use local information to predict the existence of a link between two nodes, but then supplement this local information with community membership information. For example, if two nodes, as well as some of their shared neighbors, are all in communities together, then we may infer that a link between them is more likely than if they and their shared neighbors are in different communities. We test our methods on 10 datasets from a variety of domains, ranging from scientific collaboration networks to friendship networks to gene interaction networks, and show that using community information often increases the precision accuracy of the results.

The remainder of this paper is organized as follows: First, we discuss related work, beginning with a discussion of existing link prediction methods, including local similarity-based link prediction methods, and then describe some algorithms for community detection. Next, we describe the experiments we perform and the datasets used in these experiments. After that, we discuss the results of these experiments, and finally conclude with suggestions for future work.

2 Related Work

In this section, we first discuss several methods for link prediction, beginning with those based on local similarity. We also include a discussion of how one can evaluate a link prediction method.

*Supported by AFOSR Grant FA9550-09-1-0675.

†Department of Computer Science, Cornell University

‡Department of Computer Science, Cornell University

2.1 Local Similarity Measures We consider six types of local similarity measures. Throughout this paper, we refer to these metrics as 'base metrics', as they provide the foundation for our 'enhanced metrics' that incorporate community information. For a vertex v, let $\Gamma(v)$ be the set of all neighbors of v, and let $d(v)$ be the degree of v. Then for nodes a and b, we use the following base metrics:

- Common Neighbors: $|\Gamma(a) \cap \Gamma(b)|$, the number of neighbors shared by both a and b.
- Jaccard Similarity: $\frac{|\Gamma(a) \cap \Gamma(b)|}{|\Gamma(a) \cup \Gamma(b)|}$, the number of vertices adjacent to both a and b normalized by the number of vertices adjacent to either a or b [6].
- Sorensen Similarity: $\frac{|\Gamma(a) \cap \Gamma(b)|}{d(a)+d(b)}$, the number of vertices adjacent to both a and b normalized by the sum of the degrees of a and b [20].
- Resource Allocation: $\sum_{c \in \Gamma(a) \cap \Gamma(b)} \frac{1}{d(c)}$, the sum of the inverses of the degrees of vertices adjacent to both a and b. Intuitively, this is similar to the Common Neighbors measurement, but vertices with higher degree are worth less than those with low degree, because it is assumed that a high degree vertex connected to both a and b is less meaningful than a low degree vertex connected to both a and b [23].
- Adamic-Adar: $\sum_{c \in \Gamma(a) \cap \Gamma(b)} \frac{1}{log(d(c))}$ is similar to Resource Allocation, but the denominator of the fraction is the log of the degree of the shared neighbor, rather than simply the degree [1].
- Leicht-Holme-Newman: $\frac{|\Gamma(a) \cap \Gamma(b)|}{d(a) \times d(b)}$, the number of vertices adjacent to both a and b normalized by the product of the degrees of a and b [9].

The Common Neighbors metric has been used in collaboration networks, where it has been shown that individuals who have collaborated with many of the same people are more likely to collaborate together in the future [17]. The Jaccard Similarity, Sorensen, and Leicht-Holme-Newman metrics use the same principle, but normalize this value. The Resource Allocation index is based on the principle of distributing resources along the edges of a network. If a has some unit of resource and distributes it equally to its neighbors shared by b, and each of those neighbors then allocates its portion of the resources equally between its neighbors, then the amount of resource that b receives is measured by the Resource Allocation index.

2.2 Other Link Prediction Methods We now discuss two other link prediction methods, both of which partition the network into groups and then determine node similarity based on shared group membership. These methods are inefficient and impractical for networks of more than a few thousand nodes, but are of theoretical importance. Unlike the methods in the previous section, these algorithms directly incorporate group membership information into their results.

Clauset, Moore, and Newman propose a method to determine the hierarchical structure of a network by using MCMC sampling to create a binary dendrogram that joins nodes into groups [4]. In this dendrogram, each node is initially in its own group, represented by the leaves, and then, at each internal node of the dendrogram, two groups X and Y are joined together to form a larger group. At each such step, one calculates the number of links between nodes in X and nodes in Y, and then normalizes this by the number of possible links between nodes in X and nodes in Y. This gives a value between 0 and 1 which can be interpreted as the probability that a node in X is connected to a node in Y. This process can be very slow due to the time needed to produce the dendrogram. Since this method was introduced, a variety of similar methods and models have been produced.

Another network model, the stochastic block model, partitions nodes into groups, and the probability that two nodes are connected depends only on their group memberships [22]. Each possible partition is assigned a likelihood depending on the number of links between different parts as well as the number of links within each part. Like the Clauset, Moore, and Newman hierarchical algorithm, this method is inefficient and impractical for large networks.

2.3 Evaluating a Link Prediction Method The success of a link prediction algorithm on a network N can be measured by two methods: precision and area under the receiver operating characteristic curve (AUROC) [15]. To calculate these, one can perform 10-fold cross validation. For each fold, 10% of the existing links are withheld from the network to create a new network N'. This is done 10 times, and each time, a different 10% of the links are withheld. This ensures that each link is withheld exactly once, so all links are present in the training data and the test data an equal number of times.

To measure precision, one uses N' to identify the n links that are most likely to exist in the full network, and then calculates the fraction of these n links that are present in the withheld 10% of links.

When evaluating precision scores in this paper, it

is important to note that many of the networks are incomplete (even when none of the known links are withheld). Consider a biological network in which protein interactions are identified experimentally, and only 5% of the interactions have been identified. A perfect link prediction algorithm, then, might receive a precision score of only 5%: even if all of its predictions are actually correct, only 5% of the predicted links have been confirmed and count toward the precision score. Thus, while one can compare precision scores of different algorithms against one another on the same network, they should not be compared across datasets or against an absolute standard.

To measure AUROC, one samples m pairs of edges that are not in N', where one edge e_0 in the pair is present in the list of withheld edges and the other edge e_1 is not in this list. The link prediction method is used to score each pair by determining which of e_0 or e_1 is more likely to be present in the full network. If the method determines that e_0 is more likely than e_1 to be in the full network, then the pair is assigned a score of 1. If the method determines that e_0 and e_1 are equally likely to be in the network, then the pair gets a score of 0.5, and if the method determines that e_1 is more likely than e_0 to be in the network, then the pair gets a score of 0. To estimate the AUROC, one averages these scores over all sampled pairs.

3 Community Detection Methods

We now discuss the several community detection methods that we use to identify communities in our networks.

Community detection methods have their roots in graph clustering and partitioning; as such, many community detection methods partition the network into disjoint communities. More recently, researchers have begun to develop community detection methods that identify overlapping communities. In our work, we consider algorithms of both types.

3.1 Greedy Modularity Optimization One method to determine the quality of a partition is modularity [3], which is based on the principle that a good community has many internal links but is mostly isolated from the rest of the network. The modularity of a partition is defined as follows: Let m be the number of edges in the network, $A_{i,j}$ be the number of edges between vertices i and j, k_i be the degree of node i, and $\delta(i,j)$ be 0 if i and j are in different parts of the partition and 1 if they are in the same part. Then the modularity Q of a partition is:

$$Q = \frac{1}{2m} \sum_{i,j} \left[A_{ij} - \frac{k_i k_j}{2m} \right] \delta(i,j). \tag{3.1}$$

In this paper, we use the Louvain method for greedy modularity optimization [3] to partition the network into high modularity parts.

3.2 Link Communities The next method we consider is the Link Communities method of Ahn, Bagrow, and Lehmann [2]. Unlike the Louvain method for modularity optimization, the Link Communities algorithm identifies communities that may overlap one another. This algorithm is based on the concept of placing edges, rather than nodes, into communities.

To find communities in a network N, one begins by forming a new network N' such that each node in N' represents an edge from N. Two nodes in N' are connected if the corresponding edges in N are adjacent. Edges connecting nodes in N' are weighted according to a similarity function over the edges in N, and single-linkage hierarchical clustering is used to create a dendrogram, which is then split at the point of maximum partition density. See [2] for full details.

When evaluated on networks that contain some sort of ground truth communities, this method has been shown to be more effective than others, including greedy modularity optimization, at identifying the ground truth communities [2].

3.3 Infomap The final method we discuss is the Infomap partitioning algorithm of Rosvall and Bergstrom [19]. As with greedy modularity optimization, this method produces a partitioning of the network; however, even without overlap, the results are generally of very high quality. Fortunato and Lancichinetti evaluated several detection methods and concluded that, of the methods tested, Infomap was the most accurate [8].

Inspired by the idea of a cartographer making a map, who wishes to convey important geographical information without too deep a level of detail, Rosvall and Bergstrom attempt to identify a coarse-grained representation of how information flows through a network. In a partitioning, each cluster is given a name (number) and each node within a cluster is given its own local name. The goal of the algorithm is to find a partitioning and labeling of nodes in the network so as to minimize the expected length of a random walk's description. An initial clustering is identified using a greedy algorithm, and simulated annealing is then used to improve the results.

4 Enhanced Link Prediction

In this section, we begin with an overview of our methods, and then describe the experiments that we conduct on several datasets. Specific details about our experimental methodology follow in the next section.

We perform two types of experiments. First, we modify local similarity metrics, or 'base metrics', to account for community membership. Next, we modify the hierarchical link prediction method of Clauset, Moore, and Newman using communities generated by the Link Communities method.

4.1 Modification of Local Similarity Measures

We believe that community membership information can provide valuable information for the link prediction problem. Consider the Common Neighbors similarity metric, which simply calculates the neighbors shared between two nodes. Suppose that we are using this metric to analyze a friendship network.

Suppose that individual c knows both a and b, but c knows a from the community corresponding to some school, and c knows b from the community corresponding to some workplace. Suppose additionally that a and b both have a neighbor d in common, and d knows both a and b from the community corresponding to some sports team. When calculating the probability that a and b know one another, it is possible that the shared neighbor d should count more heavily towards this probability than the shared neighbor c, since a and b both know d from the same context. The fact that a and b are in at least one community together should also count towards this probability.

With this intuition in mind, we begin with six base local similarity metrics for link prediction, and enhance each method with community information. In some metrics, we assign extra points to a pair of nodes a and b if they share neighbors in the same communities, or if a and b themselves are in the same communities.

For each method, we use the following shorthand: for nodes a and b, let $\Gamma(a)$ be the set of neighbors of a, $\Gamma(b)$ be the set of neighbors of b, $\Gamma(a,b)$ be the neighbors shared between a and b, $d(a)$ be the degree of a, and $d(b)$ be the degree of b.

As our base metrics, we consider Common Neighbors, Jaccard Similarity, Sorensen, Leicht-Holme-Newman, Adamic-Adar, and Resource Allocation, described in the Related Work section. Of these six metrics, either Common Neighbors or Resource Allocation was the most successful base metric on each of the datasets considered, and so to save space, we present only their modifications here.

We generate communities using the Louvain method for greedy modularity optimization, Infomap, and the Link Communities method. Because the Link Communities method generates communities of edges rather than nodes, we interpret its results in two ways: first, simply as a collection of overlapping communities of nodes, and second, as a partitioning of relationships. Call the former method Link Communities- Node, and the latter method Link Communities- Edge. Call the former type of community a node community, and the latter type an edge community.

We consider a variety of modifications to the original local similarity measures. For a pair of nodes (a,b), most of these modifications either assign extra points for neighbors shared between a and b that are in some of the same communities as a and b, or assign extra points for all communities that a and b are both in, or both. Some of these methods performed very poorly, whereas others were quite successful. We present the most accurate modifications below.

For each of the metrics described below, given a particular community detection method, let $C(n)$ be the set of node communities to which node n belongs. For Link Communities- Edge, let $C(n,m)$ be the set of edge communities to which edge (n,m) belongs. For greedy modularity optimization and Infomap, the size of $C(n)$ is 1, and for Link Communities- Edge, the size of $C(n,m)$ is 1. For Link Communities- Node, the size of $C(n)$ may be greater than 1.

4.1.1 Common Neighbors

We enhance the Common Neighbors (CN) base metric in 5 ways. For nodes a and b, let $CN(a,b)$ be the number of common neighbors between a and b. The first three methods ($CN1, CN2, CN3$) can be computed using the node communities obtained from greedy modularity optimization, Infomap, and Link Communities- Node. The last two methods can be computed using the edge communities from Link Communities- Edge.

- Common Neighbors 1 (CN1): In this measure, $CN1(a,b)$ begins with the base score given by $CN(a,b)$, and then for every neighbor i shared by a and b, $CN1(a,b)$ receives an additional point for every community that a, b, and i are all in.

$$CN1(a,b) = CN(a,b) + \sum_{i \in \Gamma(a,b)} |C(i) \cap C(a) \cap C(b)| . \tag{4.2}$$

- Common Neighbors Edge 1 (CNEdge1): In this measure, $CNEdge(a,b)$ begins with the base score given by $CN(a,b)$, and then for every neighbor i shared by both a and b, if the relationships between both a and i and b and i fall into the same community, $CNEdge1(a,b)$ receives another point.

$$CNEdge1(a,b) = CN(a,b) + \sum_{i \in \Gamma(a,b)} |C(i,a) \cap C(i,b)| . \tag{4.3}$$

4.1.2 Resource Allocation We enhance the Resource Allocation metric in 2 ways, the first computed using node communities from greedy modularity optimization, Infomap, or Link Communities- Node, and the second computed using edge communities from Link Communities-Edge.

- Resource Allocation 1 (RA1): $RA1(a,b)$ is the sum over all vertices $i \in \Gamma(a,b)$ of $\frac{1+|C(i)\cap C(a,b)|}{d(i)}$. This is similar to the original Resource Allocation definition, but we give extra weight to shared neighbors i that are in at least one community with both a and b, and weight i's contribution toward the total score by the number of communities that i shares with a and b.

$$RA1(a,b) = \sum_{i\in\Gamma(a,b)} \frac{1+|C(i)\cap C(a)\cap C(b)|}{d(i)}. \tag{4.4}$$

- Resource Allocation Edge 1 (RAEdge1): This is similar to $RA1$, except we give extra weight to shared neighbors i such that (i,a) and (i,b) are in at least one edge community together.

$$RAEdge1(a,b) = \sum_{i\in\Gamma(a,b)} \frac{1+|C(i,a)\cap C(i,b)|}{d(i)}. \tag{4.5}$$

4.2 Hierarchical Link Prediction We also consider a variety of link prediction methods that are based on the hierarchical link prediction algorithm of Clauset, Moore, and Newman [4]. Their method constructs a binary dendrogram representing a hierarchical community structure. This construction is extremely slow because it uses an MCMC sampling process, so we instead use the Link Communities algorithm, which also produces a hierarchical community structure. We modify Clauset, Moore, and Newman's method to work with this dendrogram. However, these results were much worse than those described in the previous section, so we do not describe them here.

5 Datasets

We test the preceding metrics, base and enhanced, on 10 datasets, described in Table 1. Networks Amazon, Email, Word, and Wiki were originally directed, and we converted them to undirected networks by simply converting every arc into an undirected edge.

6 Experimental Methodology

For each network N in Table 1, we perform experiments using 10-fold cross validation. We partition N's links into 10 equal-sized sets. We then perform 10 rounds of experiments. Each such set is used as test data in one round, while the remaining 90% of the links are used as training data. For round i, let N_i denote the network defined by the links in the training data, and let T_i denote the set of links constituting the test data.

For each round of experiments, we use the training data to generate communities using Infomap, the Louvain method for greedy modularity optimization, and Link Communities.

Then, using each of the base and enhanced metrics described in Section 4, we identify the n most likely links that are not present in N_i, where $n = \frac{|T_i|}{10}$. We then calculate how many of the n found links are actually in T_i. This value, averaged over all 10 folds, is the precision of the metric. As before, we caution that the precision scores should not be compared across networks or to an absolute standard. Many of the networks, such as the biological networks in which links are experimentally determined, are incomplete, even when links are not withheld. For such networks, even a perfect link prediction method could get a low precision score because although its predictions are all correct, the link does not exist in the known (incomplete) network. Thus, a low precision score should not be taken as an indication that a method is objectively 'bad'; rather, the precision scores should be only used to compare algorithm accuracy.

We also calculate the area under receiver operating characteristic curve. We select 1000 pairs of edges in which the first edge is present in T_i and the second edge is not present in either N_i or T_i, and then use each metric to score the two edges. Using this information and averaging over all 10 folds, we estimate the AUROC value.

7 Results

In this section, we discuss the modified local similarity metrics from section 4.1. Our results show that the modified local similarity metrics perform quite well in comparison to the base local similarity metrics. The modified hierarchical methods perform much worse than even the base local similarity metrics, so to save space we do not present the results here.

Table 1: Datasets

Network	Description	# Nodes	# Edges
Amazon [10]	A product co-purchasing network from Amazon.com Two nodes are connected if the products are frequently purchased together.	270,347	741,142
Grad [16]	Facebook network for graduate students at Rice University	503	3256
Ugrad [16]	Facebook network for undergraduate students at Rice University	1220	43208
HS [18, 14]	Gene interaction network for *H. Sapiens*	10,298	54,655
SC [18, 14]	Gene interaction network for *S. Cerevisiae*	5523	82,656
Email [5]	E-mail network from University Rovira i Virgili in Spain	1133	5452
HEP [13]	High-energy physics co-authorship network from Arxiv.com	9877	25,988
Rel [13]	General relativity physics co-authorship network from Arxiv.com	5242	14,496
Word [7]	Experimentally created word associative thesaurus Two words are connected if subjects frequently thought of one when shown the other.	23,219	305,500
Wiki [11, 12]	Administrator election network from Wikipedia.com Two nodes are connected if one participated in the election of the other.	7115	100,762

Table 2: Precision of Base Metrics

	CN	Jacc	Sor	LHN	RA	AA
Amazon	**0.3713**	0.0193	0.0193	0.0139	0.3540	0.3702
Grad	0.3713	0.5000	0.5000	0.0182	**0.7212**	0.5848
Ugrad	0.5757	0.5870	0.5870	0.0322	**0.6889**	0.6109
HS	**0.1110**	0.0113	0.0113	0.0007	0.0726	0.1042
SC	**0.1944**	0.0306	0.0306	0.0000	0.0825	0.1706
Email	**0.3509**	0.1000	0.1000	0.0018	0.3255	0.3452
HEP	0.6988	0.6288	0.6288	0.0635	**0.9227**	0.8381
Rel	0.9676	0.9828	0.9828	0.0717	**0.9903**	0.9690
Word	0.1402	0.0013	0.0013	0.0000	**0.1471**	0.1403
Wiki	**0.1773**	0.0001	0.0001	0.0000	0.1420	0.1702

7.1 Precision Table 2 shows the results of evaluating each of the base metrics on each of the networks using 10-fold cross validation. For each network, the best performing metric has been bolded. Two metrics, Common Neighbors and Resource Allocation, are each the best performing metric for half of the networks, and the other metrics are not the best for any network.

We now present the CN, $CN1$, $CNEdge1$, RA, $RA1$, and $RAEdge1$ results for all networks using all community detection methods. Table 3 contains the results for each network using communities found using the Link Communities (LC) method, the Louvain method (Mod), and Infomap (IM).

On every network except Grad, the best performing metric is one that incorporates community information. On 7 out of the 10 networks, some form of RA1 outperformed the other metrics, both base and enhanced (including ones not presented here), although it is not clear which community detection method is best. Additionally, regardless of choice of community detection algorithm, all of $CN1$, $CNEdge1$, $RA1$, and $RAEdge1$ outperform their corresponding base metrics on average, sometimes by a large factor.

Although for some networks, it appears that every metric does poorly, we again caution that this is not necessarily the case. For some networks (especially the biological networks, like HS) only a very small fraction of all links are known, and thus even a perfect link prediction would appear to have a low score. For other networks, such as Rel, it appears that the enhanced metric improves the base metric only slightly: for example, the precision of RA on Rel is 0.9903, and the precision of $RAEdge1$ is 0.9945. While this is only an increase of 0.0042 in absolute terms, it covers nearly half of the distance from the base metric score to a perfect score.

8 Analysis

In nearly every case, the enhanced metrics outperformed the associated base metrics. In the cases where the enhanced metrics performed equally to or worse than the base metrics, it is possible that the community information was flawed. To evaluate the likelihood of this possibility, we re-calculate each metric using community information obtained by applying the community detection algorithms to the entire set of edges. That is, when identifying communities, we use all edges, rather than just the 90% of edges contained in the training set. All other portions of each metric, such as the number of shared neighbors, are still calculated using only the edges in the training set. Unsurprisingly, we see a fairly significant improvement in performance: even for network Grad, where the best metric was previously the base metric RA, the best metric is now RAEdge1. Naturally, in a real application, a practitioner certainly will not have access

Table 3: Precision for Base and Enhanced Metrics

	CN	CN1(mod)	CN1(LC)	CN1(IM)	CNEdge1	RA	RA1(mod)	RA1(LC)	RA1(IM)	RAEdge1
Amazon	0.3713	0.3717	0.3220	0.3740	0.3326	0.3507	0.3519	**0.4114**	0.3783	0.4097
Grad	0.5515	0.5455	0.5364	0.5364	0.5484	**0.7212**	0.7000	0.7152	0.6879	0.7030
Ugrad	0.5757	0.6748	0.6491	0.6738	0.6671	0.6889	0.7236	0.7007	**0.7241**	0.7144
HS	0.1110	0.1196	**0.1572**	0.1203	0.1402	0.0726	0.0755	0.1111	0.0762	0.1256
SC	0.1944	0.2961	0.3746	0.2734	0.3486	0.0825	0.1520	0.3724	0.1374	**0.4707**
Email	0.3509	0.3509	0.3291	0.3418	0.3691	0.3255	0.3455	**0.3836**	0.3473	0.3764
HEP	0.6988	0.6831	0.7762	0.7031	0.8038	0.9227	0.9196	**0.9281**	0.9204	0.9277
Rel	0.9676	0.9676	0.9676	0.9676	0.9676	0.9903	0.9917	0.9917	0.9917	**0.9945**
Word	0.1402	0.1486	0.1015	0.1473	0.1101	0.1471	**0.1490**	0.1276	0.1403	0.0940
Wiki	0.1772	**0.1951**	0.1437	0.1894	0.1515	0.1420	0.1397	0.1459	0.1419	0.1187
Average	0.4139	0.4353	0.4357	0.4327	0.4439	0.4443	0.4548	0.4888	0.4546	**0.4935**

to the complete set of edges, and so must use inaccurate community information, but this experiment demonstrates the importance of correct community membership data.

In order to assist a practitioner in selecting an appropriate metric, we present the following two methods.

8.1 Metric Selection through Cross-Validation In our original experiments, we performed 10-fold cross-validation. In each iteration, 10% of the network's edges were withheld for testing and the remaining 90% used for generating communities and making predictions. For each of these training sets containing 90% of the edges from the original network, we perform another level of 10-fold cross-validation by further dividing those edges into 10 sets of training and testing edges. We apply the link prediction methods to this second layer of cross-validation sets.

For example, consider network SC. In the previous section, we created 10 sets of training edges. Call these sets SC_1, SC_2, ..., SC_{10}. For each SC_i, metric $RAEdge1$ produces some fraction improvement f_i over metric RA when evaluated on the corresponding set of test edges (the ratio of the $RAEdge1$ score to the RA score is approximately 6 on average, but varies for each SC_i). For each SC_i, we create 10 more sets of training edges, $SC_{i,1}$, ..., $SC_{i,10}$. For each $SC_{i,j}$, metric $RAEdge1$ gives some fractional improvement $f_{i,j}$ over metric RA when tested on the corresponding set of test edges. For each SC_i, we calculate the average $\overline{f_{i,j}}$ over all values j of $f_{i,j}$.

We then plot f_i against $\overline{f_{i,j}}$ for all values i, all networks, and all enhanced metrics. Results are shown in Figure 1. The purpose of this experiment is to show that $\overline{f_{i,j}}$ is strongly related to f_i:

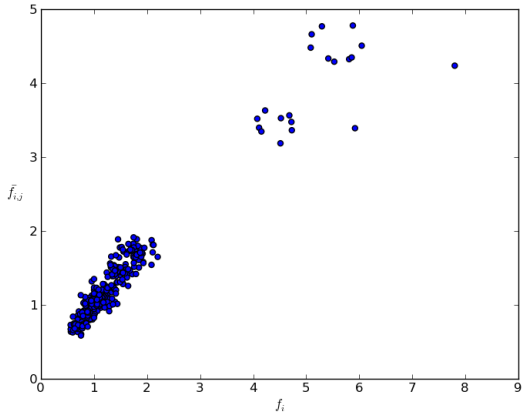

Figure 1: $\overline{f_{i,j}}$ vs. f_i

thus, to select a metric for some network, one can observe that metric's cross-validation performance on subsets of the network.

We see that there is a fairly strong relationship between each f_i and $\overline{f_{i,j}}$. In particular, when f_i is extremely high (as in the case of network SC), $\overline{f_{i,j}}$ is also very high, and vice versa.

These results give guidance to users looking to apply these methods to real data. Although no metric is best for every network, one can simply perform cross-validation on that network to determine whether a particular metric is likely to be successful when applied to the complete data.

8.2 Metric Selection through Comparison of Existing Edges and Non-Edges In addition to performing the cross-validation method of metric selection described above, one can also use the existing data to compare pairs of nodes that are connected and pairs of nodes that are

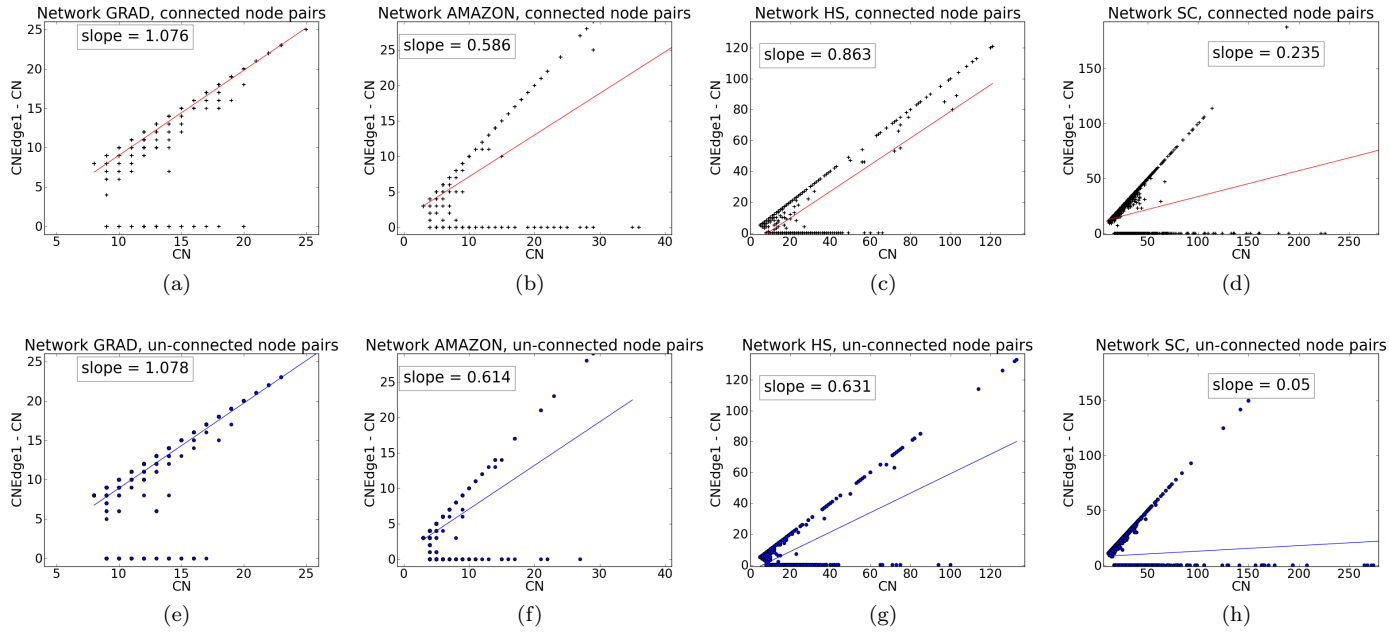

Figure 2: Plots of $CNEdge1 - CN$ vs. CN. The top row contains values for pairs of nodes that are connected, and the bottom row contains values for pairs of nodes that are not connected.

not connected. In this experiment, for every training network, we generate a list of all pairs of nodes (u, v) that share at least one neighbor. For some of these pairs, u and v are connected, and for other pairs, they are not. For each pair in the list, we calculate the value of the $CN(u, v)$ and $CNEdge1(u, v)$ metrics. We then plot $CNEdge1(u, v) - CN(u, v)$ vs. $CN(u, v)$, and determine whether the $CNEdge1(u, v) - CN(u, v)$ values differ significantly between pairs of nodes that are connected and not connected, that have the same $CN(u, v)$ value.

On Grad and Amazon, $CNEdge1$ performed worse than CN, and on SC and HS, $CNEdge1$ outperformed CN. To save space, we present the plots for only these networks in Figure 2. In Figure 2, the top row contains the plots for connected pairs of nodes, and the bottom row contains the plots for un-connected pairs of nodes. We limit the range of CN and $CNEdge1 - CN$ values that we consider, and consider only those pairs of nodes that had high CN or $CNEdge1$ values, because these pairs of nodes are the ones that effect precision scores (as precision only considers the most likely edges). In this case, we consider the top e pairs of nodes as measured by either CN or $CNEdge1$, where e is the number of edges in the test set. For each plot, we calculate the best-fit line.

From this figure, we quickly reach several conclusions. First, for those networks (SC and HS) on which $CNEdge1$ outperformed CN, the slope of the best-fit line is significantly higher for the connected nodes than for the un-connected nodes (this is especially true for SC). This indicates that connected pairs of nodes tend to have much higher $CNEdge1$ values than un-connected nodes with the same CN value, and so $CNEdge1$ does a good job in discriminating between connected and un-connected node pairs. In contrast, on Grad and Amazon, the slopes of the best-fit lines are nearly identical. This strongly suggests that, for these two networks, $CNEdge1$ is unlikely to outperform CN (as is indeed the case).

Note also that for HS and SC, there are relatively few pairs of connected nodes that have very high CN values and low $CNEdge1 - CN$ values (along the x-axis). This is particularly apparent for HS, which has no pairs of connected nodes along the x-axis with CN greater than approximately 70. In contrast, there are many pairs of un-connected nodes in this location. Conversely, for Amazon and Grad, the plot containing connected pairs has many elements with high CN and $CNEdge1 - CN = 0$.

This method of comparing plots is a simple way to determine whether the incorporation of community

information is likely to be useful, and unlike the earlier cross-validation method, does not require reapplication of community detection methods.

9 Conclusion and Future Work

We have showed that enhanced local similarity metrics often outperform their associated base metrics. Averaged over all 10 networks, enhanced metrics $CN1$, $CNEdge1$, $RA1$, and $RAEdge1$ had higher precision scores than their corresponding base metrics, and similar AUROC scores. Although no single metric was best for every network, we showed that one can perform cross-validation or compare metric values for existing pairs of connected and un-connected nodes to determine whether a particular method is likely to succeed on a given network.

Future directions for this problem might focus on other ways to determine which community detection method is best suited for a particular network. This problem is naturally related to the problem of determining which community detection method is best suited for finding communities within a network. A related problem is that of determining which base local similarity metric is best for a particular network. Can one determine which metric to use based on features of the network?

References

[1] L. Adamic, and E. Adar. Friends and neighbors on the web. *Social Networks*, 25(3), 2003.

[2] Y. Y. Ahn, J. P. Bagrow, and S. Lehmann. Link communities reveal multiscale complexity in networks. *Nature*, 466, 2010.

[3] V. D. Blondel, J.-L. Guillaume, R. Lambiotte, and E. Lefebvre. Fast unfolding of communities in large networks. *Journal of Statistical Mechanics: Theory and Experiment*, 2008(10), 2008.

[4] A. Clauset, C. Moore, and M. Newman. Hierarchical structure and the prediction of missing links in networks. *Nature*, 453, 2008.

[5] R. Guimera, L. Danon, A. Diaz-Guilera, F. Giralt, and A. Arenas. Self-similar community structure in a network of human interactions. *Physical Review E*, 68, 2003.

[6] P. Jaccard. Etude comparative de la distribution florale dans une portion des alpes et des jura. *Bulletin de la Societe Vaudoise des Science Naturelles*, 37, 1901.

[7] G. Kiss, C. Armstrong, R. Milroy, and J. Piper. An associative thesaurus of english and its computer analysis. In A. Aitken, R. Bailey, and N. Hamilton-Smith, editors, *The Computer and Literary Studies*. Edinburgh: University Press, 1973.

[8] A. Lancichinetti and S. Fortunato. Community detection algorithms: A comparative analysis. *Physical Review E*, 80(5), November 2009.

[9] E. A. Leicht, P. Holme, and M. Newman. Vertex similarity in networks. *Phys. Rev. E*, 73, 2006.

[10] J. Leskovec, L. Adamic, and B. Adamic. The dynamics of viral marketing. *ACM Transactions on the Web*, 1(1), 2007.

[11] J. Leskovec, D. Huttenlocher, and J. Kleinberg. Signed networks in social media. *CHI*, 2010.

[12] J. Leskovec, D. Huttenlocher, and J. Kleinberg. Predicting positive and negative links in online social networks. *WWW*, 2010.

[13] J. Leskovec, J. Kleinberg, and C. Faloutsos. Graph evolution: Densification and shrinking diameters. *ACM Transactions on Knowledge Discovery from Data*, 2007.

[14] C. Liao, K. Lu, M. Baym, R. Singh, and B. Berger. Isorankn: Spectral methods for global alignment of multiple protein networks. *Bioinformatics*, 25, 2009.

[15] L. Lu and T. Zhou. Link prediction in complex networks: a survey. *Physica A: Statistical Mechanics and its Applications*, 390(6):1150–1170, March 2011.

[16] A. Mislove, B. Viswanath, K. Gummadi, and P. Druschel. You are who you know: inferring user profiles in online social networks. *WSDM*, 2010.

[17] M. E. J. Newman. Clustering and preferential attachment in growing networks. *Phys. Rev. E*, 64, 2001.

[18] D. Park, R. Singh, M. Baym, C. Liao, and B. Berger. Isobase: A database of functionally related proteins across ppi networks. *Nucleic Acids Research*, 2011.

[19] M. Rosvall and C. T. Bergstrom. Maps of information flow reveal community structure in complex networks. *PNAS*, 105(4):1118–1123, January 2008.

[20] T. Sorensen. A method of establishing groups of equal amplitude in plant sociology based on similarity of species content and its application to analyses of the vegetation on Danish commons. *Biol. Skr.*, 5, 1948.

[21] M. P. H. Stumpf, T. Thorne, E. de Silva, R. Stewart, H. J. An, M. Lappe, and C. Wiuf. Estimating the size of the human interactome. *PNAS*, 105, 2008.

[22] H. White, S. Boorman, and R. Breiger. Social structure from multiple networks: blockmodels of roles and positions. *Am. J. Sociol*, 81, 1976.

[23] T. Zhou, L. Lu, and Y.-C. Zhang. Predicting missing links via local information. *Eur. Phys. J. B*, 71, 2009.