ResearchGate

See discussions, stats, and author profiles for this publication at: https://www.researchgate.net/publication/228665526

Node roles and community structure in networks

Article · August 2007

DOI: 10.1145/1348549.1348553

CITATIONS
79

READS
706

3 authors, including:

Jerry Scripps
Grand Valley State University
32 PUBLICATIONS **470** CITATIONS

SEE PROFILE

Abdol-Hossein Esfahanian
Michigan State University
99 PUBLICATIONS **2,817** CITATIONS

SEE PROFILE

All content following this page was uploaded by Abdol-Hossein Esfahanian on 21 May 2014.

The user has requested enhancement of the downloaded file.

Node Roles and Community Structure in Networks

Jerry Scripps
Computer Science and Engineering
Michigan State University
E. Lansing, MI
scripps@msu.edu

Pang-Ning Tan
Computer Science and Engineering
Michigan State University
E. Lansing, MI
ptan@msu.edu

Abdol-Hossein Esfahanian
Computer Science and Engineering
Michigan State University
E. Lansing, MI
esfahanian@cse.msu.edu

ABSTRACT

A node role is a subjective characterization of the part it plays in a network structure. Knowing the role of a node is important for many link mining applications. For example, in Web search, nodes that are deemed to be authorities on a given topic are often found to be most relevant to the user's queries. There are a number of metrics that can be used to assign roles to individual nodes in a network, including degree, closeness, and betweenness. None of these metrics, however, take into account the community structure that underlies the network. In this paper we define community-based roles that the nodes can assume (ambassadors, big fish, loners, and bridges) and show how existing link mining techniques can be improved by knowledge of such roles. A new community-based metric is introduced for estimating the number of communities linked to a node. Using this metric and a modification of degree, we show how to assign community-based roles to the nodes. We also illustrate the benefits of knowing the community-based node roles in the context of link-based classification and influence maximization.

1. INTRODUCTION

A network consists of nodes connected by directed or undirected links. It is used to represent complex, relational data such as web pages or social networks. The nodes can be assigned roles, which are subjective characterization of the part they play in the network. For example, within the web, an authoritative page is one that is referred to by many other pages whereas a hub page is one that has hyperlinks to many other pages.

There are a number of metrics that can be used to determine the roles of individual nodes in a network. Among those most widely used are degree, closeness, betweenness, and rank. Degree can be used to assess a node's popularity while closeness and betweenness can be used to assess its centrality. Rank, such as that used in the HITS [6] or PageRank [9] algorithms are measures of authority within a network. Knowledge of the node role (popularity, centrality, authority) is useful for many link mining applications such as Web search, threat detection, and co-citation analysis.

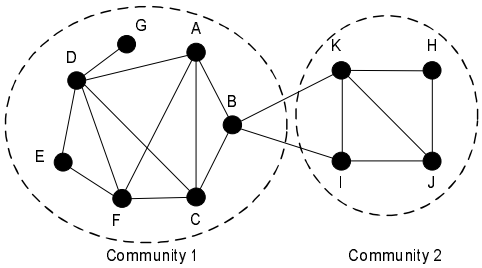

Figure 1: Groups within a network

Network community refers to groups of nodes that share similar properties. Despite its importance, none of the metrics that are used to define node roles explicitly use the community concept. Knowing the role that a node assumes with respect to its related communities would be a new and valuable tool for analysts. For example, in threat detection and crime analysis, knowing that a person has contacts with many groups could be valuable information.

In this paper, we define the various community-based roles a node can assume and show how existing link mining techniques may benefit from the knowledge of such roles. For example, a node whose role is defined as an *ambassador* has links to many nodes from different communities while another node whose role is defined as a *big fish* has links only to other nodes in the same community. We offer two examples to illustrate the advantages of assigning community-based roles to nodes, namely, *influence maximization* and *link-based classification*.

The problem of influence maximization can be thought of as finding the best k people to target in order to maximize the number of people that will eventually be influenced (e.g. adopt an idea, buy a product, etc). Links are assigned a weight between 0 and 1 representing the probability that one node influences another when it is activated. Several algorithms [3, 5] have been developed in recent years to identify the most promising set of nodes to activate. These algorithms however focus only on maximizing the number of activated nodes at the end of the influence diffusion process. In some cases, it may be more useful to maximize the number of communities that are influenced. As an example, a marketer might be interested in not only informing as many people as possible about their product but might also wish

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.
Joint 9th WEBKDD and 1st SNA-KDD Workshop '07 (WebKDD/SNA-KDD'07) August 12, 2007 , San Jose, California , USA
Copyright 2007 ACM 978-1-59593-848-0 ...$5.00.

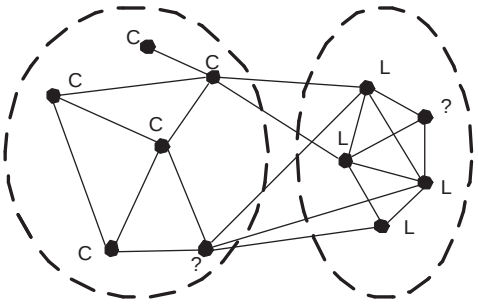

Figure 2: Classifying related objects

to maximize their reach to different demographic groups.

Figure 1 shows a small network of eleven nodes from two communities—nodes A-G belong to community 1 and nodes H-K belong to community 2. Suppose that we wish to find the one best node in this network to maximize the spread of influence. Current algorithms would choose to activate node D. Depending on the influence diffusion model and weights of the links, activating node D may not influence any of the nodes in community 2. Choosing node B, on the other hand, would elevate the chances that nodes in both communities are influenced.

Link-based classification is the task of categorizing nodes using the node features and its link information. To illustrate this task, consider the classification problem of predicting the political leaning of a person, either conservative (C) or liberal (L), given the network of people shown in Figure 2. The training examples are nodes labeled as C or L and the test examples are nodes labeled as ?. Links between the people represent friendship and the dashed ovals indicate the two communities. Several studies [2, 7, 14] have shown that the performance of traditional classifiers can be improved by using link information—specifically the class of the neighboring nodes. In Figure 2, one of the test examples is friends with only people in its community while the other has friends in both. It is likely that using the classes of neighbors would be more effective with the former test example than the latter. Knowledge of community-based roles could be helpful to link-based classifiers in deciding when to use information about a node's neighbor.

Our proposed community-based node roles can be easily defined when the membership information of the communities is available. Otherwise, a metric is needed to estimate the number of communities related to each node in the network. We have therefore introduced a new metric called *rawComm* that gives a reasonably accurate measure of community belongingness, assuming that the links in the network strongly support the community structure. Even when the assumption only weakly holds, our experimental results show that it still provides enough useful information to correctly assign community-based roles to nodes. Finally, we have conducted extensive experiments to demonstrate the benefits of knowing the community-based node roles in the context of link-based classification and influence maximization.

2. RELATED WORK

A network is an interconnected set of nodes related to each other by links that can have weights associated with them. We will only be considering undirectional, unweighted links in this study. Recent research has several categories of networks. A regular network is one where all of the nodes have a link to a fixed number of other nodes. A random network is one where the links between the nodes are completely random. Small world networks [13] are somewhere between regular and random networks. They are characterized by many small groups of tightly connected nodes (like regular networks) with a few random links that connect the small groups (like random networks). Because of this, small world networks have the property that every two nodes are connected by a relatively short path. Scale-free networks [1] have the property that the degrees of the nodes must follow a power law function; the probability $pr(k)$ that a node will have k neighbors is proportional to k^{-y} where y is usually between 2 and 3.

2.1 Node Roles

Role is a concept that is used to describe the behavior of a node in relationship to its neighbors and to the network at large. The discipline of social network analysis contains several centrality measures used to determine the roles of nodes in a network. Of these the most prominent are degree, closeness and betweenness [12]. Degree is the sum of the links attached to a node, $C_D(n_i) = \sum_j I\big[(i,j) \in E\big]$ where I is a 0/1 indicator function.

Closeness is the reciprocal of the sum of all the geodesic (shortest) distances from a given node to all others, $C_C(n_i) = \Big[\sum_{j=1}^{N} d(n_i, n_j)\Big]^{-1}$ where $d(u,v)$ is the geodesic distance from u to v. Nodes with a small C_C score are closer to the center of the network while those with higher scores are closer to the edge. Betweenness, another metric that measures how centrally located a node is, is defined as $C_B(n_i) = \sum_{j<k} \frac{g_{jk}(n_i)}{g_{jk}}$ where g_{jk} is the number of geodesic paths from j to k and $g_{jk}(n_i)$ is the number of geodesic paths from j to k that go through i. A higher betweenness value for a node means that it is on more shortest-paths between nodes, which is an indication of the node's importance.

All three of these metrics have values greater than zero with an upper bound based on the size of the network. All three can be normalized to a value between 0 and 1 by dividing it with its maximum value.

2.2 Communities

An assumption of networks as described above, is that there are communities of nodes which are not explicitly exposed but that the links infer. In social networks we think of friends, family, and colleagues as forming communities. Data mining and link mining techniques have largely ignored utilizing this potentially helpful knowledge.

Even though there has not been an effort to exploit this knowledge there have been many clustering and community finding algorithms proposed for linked data (for a review see [4]). Some are global while others find a community for a small set of nodes. They can also be overlapping or disjoint. All of them use the links to form the communities, that is, they implicitly assume that the links provide evidence for the communities.

The number and diversity of the proposed community finding techniques suggests that there is no agreed-upon best technique which makes it difficult to explicitly use community knowledge.

3. COMMUNITY-BASED NODE ROLES

In this section we will define the community-based roles. We then introduce a novel approach to measuring the compatibility between communities and the link structure of a network. Finally we present a new metric that we use to estimate the number of communities to which a node belongs.

3.1 Community Metric

Our measure of community assumes that a community is defined by a clique (maximal complete subgraph) in a network. So a group that forms a clique will be considered one community; another group that forms two non-overlapping cliques will be considered two communities. Consider a clique of 5 nodes and remove one of the edges: it is no longer one community but we would not think of it as two either. We are looking for a metric that will assign a community value to such a group of slightly more than one. A group that forms two non-overlapping cliques except one connecting edge should have a score of slightly less than two.

It should be noted that the clustering coefficient used by Watts and Strogatz [13] to identify small world networks does give a measure of a components' *cliqueness* but for a given node it measures the ratio of actual edges within its immediate neighborhood to the total possible edges. We are looking for a metric that approximates the number of communities that a nodes' neighbors form.

Our metric, which we call rawComm is to be an approximate measure of the number of communities to which a node is attached. We assume that the communities are hidden but that links provide evidence of community. We define p as the probability that two linked nodes are in the same community and q as the probability that two non-linked nodes are in different communities. By using the probabilities p and q in our definition the metric becomes more flexible and is therefore useful for communities defined by means other than cliques.

Given a network $G = (V, E)$ without the community assignments the values of p and q will not be known but approximations can be calculated by sampling or finding a similar network that does have community assignments. In a previous work [10] we defined incomplete edges as edges (links) that connect two nodes in different clusters (communities) and impure edges as non-links that appear within a cluster (community). The approximation for p would then be the complete links divided by the total number of links and for q the number of pure links divided by the total number of non-links:

$$p = \frac{\text{Complete node pairs}}{\text{Total linked node pairs}}$$

$$q = \frac{\text{Pure node pairs}}{\text{Total non-linked node pairs}}$$

Our metric rawComm is defined as:

$$rawComm(u) = \sum_{v \in N(u)} \tau_u(v)$$

where $N(u)$ is the neighborhood of u — that is all of the nodes that are directly linked to u — and $\tau_u(v)$ is

$$\tau_u(v_i) = \frac{1}{1 + \sum_{v_j \in N(u)} I(v_i, v_j) \cdot p + \bar{I}(v_i, v_j) \cdot (1 - q)}$$

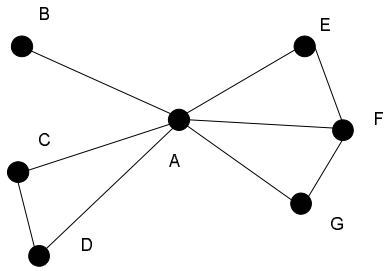

Figure 3: Calculating τ

$I(x, y)$ is an indicator function that is 1 if there is a link between x and y and 0 otherwise. $\bar{I}$ is 1 if there is not a link and 0 otherwise. It is convenient to think of $\tau_u(v)$ as the contribution of node v to node u's rawComm score.

Node v's contribution to u's rawComm community score depends on how connected v is to the nodes in u's neighborhood. The denominator in the definition of τ is the expected number of other nodes in u's neighborhood are in a community with v_i. The 1 represents the node v_i itself and the expression $I(v_i, v_j) \cdot \mathrm{p} + \bar{I}(v_i, v_j) \cdot (1 - q)$ is the probability of v_i and v_j being in the same community. Taking the reciprocal of the expectation then is the contribution of node v_i to u's rawComm score.

EXAMPLE 1: *Refer to the network in Figure 3 where we will assume that $p = 1$ and $q = 1$. Node B is connected to A but not to any others so $\tau_A(B) = 1$ which essentially adds one community to A's rawComm score. $\tau_A(C) = \tau_A(D) = 1/2$ so their sum also adds one community to A's rawComm score. So far the calculation is straightforward.*

Looking at node E we see that it is connected to only one other node with A so $\tau_A(E) = 1/2$. Likewise $\tau_A(G) = 1/2$. Since F is connected to two other nodes with A, $\tau_A(F) = 1/3$. So the sum of E, F and G's scores is 4/3. We interpret this to mean that those three nodes appear to be part of the same community but because E and G are not connected we are not sure - it could be more than one so their sum is slightly higher than one. Note that if E and G were connected the four would form a clique and the sum of their scores would be exactly one.

In general a group of nodes that form a clique will have τs that sum up to p. If they are connected but not completely, as the number of missing links grows the contribution of their scores also grows. The interpretation is that as groups of nodes become less densely connected the higher the probability that they form multiple communities which is the property discussed in the beginning of this section.

3.2 Community-Based Roles

We define the community-based role of a node according to the number of communities and links incident to it. Figure 4 shows a community-degree chart that is divided into four quadrants for the four different roles. The vertical axis represents the degree while the horizontal axis represents the community metric.

The community-based node role is identified based on which of the four quadrants a node falls into. Nodes in the upper right quadrant are those with a high degree and a high community score. They act as *ambassadors*, providing connections to many different communities. The upper left quadrant contains what we call *big fish* from the cliche "big

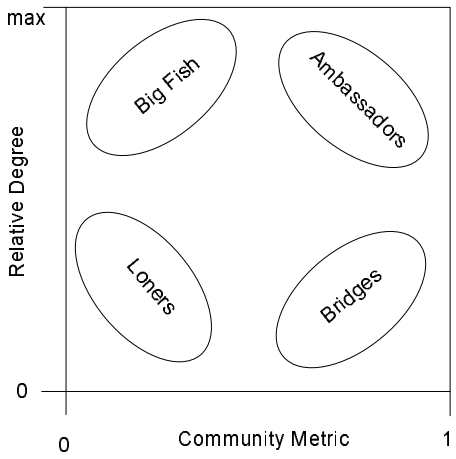

Figure 4: Community-degree chart

fish in a small pond" meaning that they are very important only within a community. This is due to their having a high degree but a relatively small community score. In the lower right quadrant are those with a low degree but a high community score. These we call *bridges* because they serve as bridges between a small number of communities. Finally, in the lower left are the *loners*—those with a low relative degree and low community score.

The metrics shown in the community-degree chart have been normalized to values between 0 and 1. For the community metric, we subtracted the minimum and divided by the range between maximum and minimum. For degree, we divided by the highest degree node in the network, giving us a relative degree score between 0 and 1. In our experiments, we chose a threshold of .5 to classify the node roles; however, depending on the distributions of degree and community metric scores, other thresholds can be chosen.

3.3 Beyond the Immediate Neighborhood

In our definition of rawComm we consider only the effect of the nodes in the immediate neighborhood of a node. While using information beyond the immediate neighborhood might improve the accuracy of rawComm it would be much more computationally expensive. For this paper we used a normalized rawComm primarily to define roles. It is important to have a relatively good estimate but absolute accuracy is not critical. If a particular node's score is slightly off, it is still given the proper role designation then rawComm has accomplished its task. The important aspect of commPct is that it is proportionally accurate — that is a node with a high probability of being attached to many communities will have a correspondingly high commPct score.

3.4 Analysis of Algorithm

The runtime complexity for rawComm for a single node is $O(\delta^2)$ where δ is the maximum degree of the network. So calculating it for all the nodes is $O(n\delta^2)$ where n is the number of nodes. The space requirements are $O(n^2)$ which is the size of the network.

This compares favorably to other methods of finding communities. The complexity of determining the community membership given the network structure is $O(nk)$. Using a community finding algorithm such as the Normalized Cut method from Shi and Malik [11] can be much greater since it involves finding the eigenvalues of the network.

4. APPLICATIONS OF COMMUNITY-BASED ROLES

Community-based roles can be useful in a number of ways. Just by themselves these roles can provide useful information to analysts in areas as such as anti-terrorism and law enforcement. In searching for potential terrorist threats, for example, analysts may find it useful to concentrate on suspects with certain roles. If they were looking for persons with few friends but having diverse contacts they could focus on bridges.

Community-based roles could also be utilized in existing techniques. The area of link mining has a number of techniques that use the relationships between objects to rank objects, select influential nodes, find communities as well other other tasks. Many of them could potentially benefit from knowledge of the objects' community role. We will discuss the two techniques of influence maximization and link-based classification.

4.1 Influence Maximization

As described in the introduction, influence maximization is concerned with finding the most influential nodes in a network. We assume that the nodes in the network are capable of adopting an idea, purchasing a product or something similar. This process is referred to as activating. We also assume that nodes that are activated have the ability to influence their immediate neighbors who themselves may choose to activate. The problem becomes choosing the best nodes to initially activate in order to maximize the number of activated nodes at the end of the process.

In the paper by Kempe, et al [5] several models are introduced that describe the behavior of the node activation. In our experiments we chose to use the Independent Cascade model. Under this model, influence is spread from node to node in discrete steps. A node i that becomes active in step t has one chance to make his inactive neighbors active in step $t+1$. The probability that node i will activate node j in their paper will be called the edge weight.

The work in this area is exclusively concerned with maximizing only the raw number of nodes activated. However, we propose extending the problem to focus on the number of communities covered. A community is covered if one of the nodes in the community is activated. Our approach will be to choose the initial set of nodes using the community-based node roles in order to maximize the groups covered. The results of our experiments will show that using roles to maximize group coverage shows improvement over the other influence maximization methods.

4.2 Link-Based Classification

Link-based classification uses both the attribute data from the objects as well as data acquired using the links. Previous studies have shown that ordinary classification can be improved by using linked data. Chakrabarti, et al.[2] have shown that using linked data can be helpful in some circumstances. In their paper, the authors were able to show that in some circumstances using the data from neighbors is not helpful but using the class from neighbors can be. A study by Yang, et al.[14], shows that data sets can contain

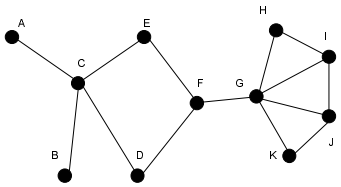

Figure 5: Sample Network

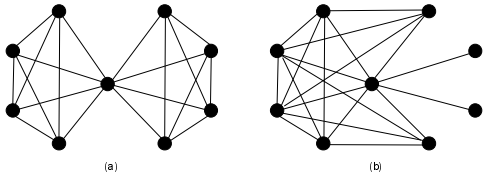

Figure 6: Comparison of rawComm to Clustering Coefficient

different types of regularities. For example with encyclopedia regularity nodes tend to link to nodes of the same class. They conclude that using the class of neighbors is helpful for some data sets while in others it is not - it depends upon the regularities present.

Since different data sets contain different regularities we contend that different regularities could exist in the same collection. It could be possible that some nodes can benefit from knowing the class of their neighbors while other nodes would not benefit. To improve the classifier under these conditions requires knowing which nodes should make use of their neighbor's class.

We propose to use our community-based roles to decide which nodes would benefit from its neighbor's class. We hypothesized that big fish and loners that have a low community metric score are more likely to be conformists and therefore are more likely to have the same class as their neighbors. Conversely, bridges and ambassadors who are connected to many groups are more likely to be independent. We propose that using the neighbor's class for loners and big fish and not for ambassadors and bridges will improve the accuracy of a link-based classifier. In the experiments section we show results that support this proposition.

5. EXPERIMENTAL EVALUATION

The purpose of this section is to provide the results of experiments which will demonstrate the distinctiveness and utility of rawComm. Specifically we show that:

- rawComm provides community information about a node that is not available from other metrics.
- rawComm is a proportionally accurate measure of the number of communities to which a node belongs.
- the accuracy of rawComm is relative to the extent to which the community structure aligns with the link structure of the network.
- the community-based role nodes follow a fairly predictable distribution

5.1 Distinctiveness of rawComm

Since rawComm is a new metric it is instructive to illustrate how it differs from the other metrics in terms of community information. Looking at Figure 5 we will show that different nodes will have high scores for the different metrics. Table 1 lists the values for degree, closeness, betweenness centrality as well as clustering coefficient, rawComm (we used $p = q = 1$ for simplicity) and commPct. Notice that the low rawComm is 1 and the high is 4.

The degree metric appears to have many of the same values as rawComm but notice that for node G which appears

Table 1: Comparison of Metrics

Node	C_D	C_C	C_B	clust Coef	raw Comm	comm Pct
A	1	3.88	0.00	1.00	1.00	0.00
B	1	3.88	0.00	1.00	1.00	0.00
C	4	5.83	17.50	0.40	4.00	1.00
D	2	5.33	9.00	0.67	2.00	0.33
E	2	5.33	9.00	0.67	2.00	0.33
F	3	6.17	25.50	0.50	3.00	0.67
G	5	6.83	24.00	0.53	2.67	0.56
H	2	4.82	0.00	1.00	1.00	0.00
I	3	5.32	0.00	1.00	1.33	0.11
J	3	5.32	0.00	1.00	1.33	0.11
K	2	4.82	0.00	1.00	1.00	0.00

to be connected to two communities, degree is 5 but rawComm is 2.67. Closeness and betweenness like degree correlate somewhat with rawComm but by comparing nodes F and G it is obvious that closeness does not capture the community knowledge. We can see that betweenness also fails to capture this knowledge by comparing node C (4 communities) with node G (2 communities).

Although clustering coefficient and rawComm are somewhat (negatively) related rawComm will always give a better estimate of the number of communities. This is shown in Figure 6 where the center node in both (a) and (b) have the same clustering coefficient it appears that in (a) the node belongs to two communities, whereas in (b) it appears to belong to three. This is borne out by the rawComm values of 2 for (a) and 3.23 for (b). It should be clear by now that commPct is a unique metric.

5.2 Comparison to community-finding algorithm

In this section we will compare the rawComm metric to actually finding the communities. Normalized cut (Ncut) [11] is a graph segmenting algorithm from the family of spectral partitioning methods which has gained much attention recently for its ability to globally optimize partitions. We will use this algorithm to find communities for a comparison to our rawComm metric.

Since the algorithm requires that the number of partitions (or communities) is given as a parameter we ran on every possible number of communities from 2 to $n - 1$. We compared the algorithms on the FaceBook data as well as many of the sets from uci-net. We show the results below from a typical set, karate, which represents the 78 relationships between the 34 students of a karate studio. In each iteration we used Ncut to find k communities and then for each node we determined the actual number of communities to which it was connected. We could then compare that to the number of communities predicted by rawComm. We calculated

rawComm two ways: first by using $p = q = 1$ for a baseline and second by calculating the p and q from the communities that were found using Ncut. Notice that when the number of communities is small many of the edges will be within the communities as will many of the non-edges, so p will be high and q will be low. When the number of communities is large p will be low and q will high.

Table 2: rawComm vs. Ncut communities

Nbr of groups	Degree SSE	p=q=1 SSE	p,q est. SSE	Q value
2	797.00	106.21	5.52	0.22
3	753.00	89.13	5.99	0.22
4	627.00	53.50	6.68	0.24
5	573.00	41.80	8.79	0.21
6	469.00	28.88	8.66	0.14
7	415.00	35.98	15.96	0.12
8	343.00	46.57	28.97	0.11
9	307.00	51.92	26.07	0.05
10	291.00	58.67	32.88	0.09
11	242.00	58.97	26.35	0.06
12	227.00	63.32	28.69	0.08
13	173.00	101.15	31.09	0.01
14	151.00	110.28	32.10	0.01
15	133.00	123.42	32.32	-0.01
16	128.00	101.42	22.55	-0.04
17	84.00	138.60	19.48	-0.03
18	115.00	108.30	20.49	-0.05
19	64.00	172.54	22.27	-0.04
20	91.00	140.39	21.19	-0.03
21	57.00	172.71	13.92	-0.10
22	32.00	227.27	22.47	-0.07
23	25.00	232.73	16.46	-0.16
24	22.00	245.81	26.18	-0.07
25	22.00	246.46	27.26	-0.13
26	22.00	246.81	18.67	-0.15
27	6.00	314.05	18.56	-0.15
28	10.00	279.63	11.32	-0.19
29	10.00	279.91	7.98	-0.17
30	4.00	319.51	5.08	-0.18
31	5.00	311.65	3.93	-0.19
32	2.00	346.88	4.28	-0.20
33	4.00	332.56	3.18	-0.22

In order to get a measure for how well rawComm compared to the actual number of communities we calculated a sum squared error (SSE) statistic by summing up, for all nodes, the square of the difference between the actual number of communities and the raw comm. We also calculated the SSE for the difference of the degree and the number of communities.

The last column, labeled Q, is a modularity measure proposed by [8]. It is the fraction of links within a community minus the expected value of the same fraction for a random graph. So a score of zero means a poor community grouping while better groupings have numbers farther from zero.

Looking at Table 2 we can see that the SSE for rawComm (columns 3 and 4) is much lower than for degree which is not unexpected. As the number of communities grows degree becomes a better predictor which is also not unexpected. Imagine partitioning the nodes into n clusters - each node in its own cluster. Degree would be a perfect predictor of community adjacency. However, such small communities are often of little value. Looking at the column under Q it is obvious that the more natural community structures are in the range of 2 to 8.

One of the key findings can be seen from comparing columns 3 where we use $p = q = 1$ and 4 where we estimate p and q from the communities that are found from Ncut. When p and q are known or accurately predicted the improvement is dramatic. All of the SSE values from column 4 are less than 34, the number of nodes which indicates that for any given node, on average, our error is less than 1, a most encouraging result.

Even though the scores for rawComm are much better when a good estimate of p and q are given, when $p = q = 1$ is used the prediction is still not too bad. Estimating p and q are domain specific tasks, beyond the scope of this paper but if an estimate can be made it will almost certainly improve the results.

5.3 Effect of p and q on Accuracy

We have seen how rawComm can estimate community belongingness in the previous section, now we will show the accuracy of that prediction is based on the alignment between the community and link structures. We use p and q to measure this alignment: a high value of p means that most links are within communities and a high value of q means that the non-links are between communities - this represents a good alignment. Lower values of p and q mean a worse alignment. To see how rawComm is affected by different values of p and q refer to Figure 7. In (a) the network has twenty nodes that are grouped into 5 hidden communities and having the links shown. In (b) the same nodes are grouped in the same communities but the links are different. In this example even though we know the communities we assume that our algorithm for calculating rawComm does not. We calculate $p = 31/39$ and $q = 1$ in the top network and $p = 20/34$ and $q = 145/156$ in the bottom network.

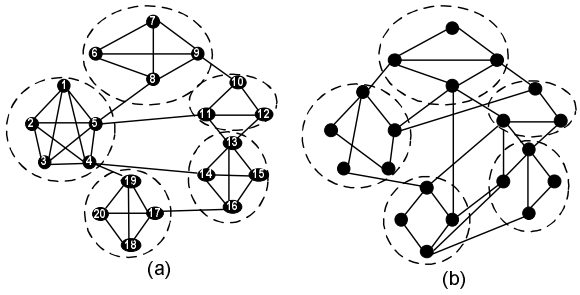

Figure 7: Effect of p on τ

Table 3 shows the actual community membership numbers versus the rawComm estimation. The correlation coefficients are .957 for network (a) and .798 for network (b) while the sum squared error (SSE) is .89 for (a) and 7.5 for (b) [1]. Thus, as p and q get progressively smaller rawComm's estimation gets progressively worse. This is not unexpected

[1]Though correlation is a popular statistic we will use the SSE for comparing rawComm to actual communities because it captures the absolute difference between the values rather than just the relative difference

Table 3: Comparison of rawComm to actual

	Network a		Network b	
Node	Actual	rawComm	Actual	rawComm
1	1.0	1.14	2.0	3.30
2	1.0	1.14	1.0	1.87
3	1.0	1.14	2.0	1.87
4	3.0	3.14	1.0	1.87
5	3.0	3.14	3.0	3.30
6	1.0	1.12	2.0	2.43
7	1.0	1.12	1.0	1.26
8	2.0	2.12	4.0	3.45
9	2.0	2.12	2.0	2.43
10	2.0	2.09	3.0	2.63
11	3.0	2.46	4.0	3.10
12	2.0	1.46	2.0	2.08
13	2.0	2.21	2.0	2.65
14	2.0	2.12	3.0	2.31
15	1.0	1.12	1.0	1.26
16	2.0	2.12	2.0	2.08
17	2.0	2.12	3.0	3.30
18	1.0	1.12	1.0	1.87
19	2.0	2.12	3.0	2.81
20	1.0	1.12	2.0	2.81

though, since as p and q get smaller the links do not really provide very good evidence of community belonging. One can easily inspect the networks to see that the nodes in network (a) appear to fit more naturally into the communities than those of network (b).

5.4 Role Distribution

So far we have described the metrics and introduced the chart that separates the node types. In this section we plot the node roles for two real data sets so that we can get a clearer picture of how the nodes in a network will be distributed in our chart.

Generally we expect to see a corridor of nodes between the lower left corner where both degreePct and commPct are zero to the upper right corner where they are both 1. The rationale for this is that nodes with a high degree are more likely to have a high rawComm. For any given node the minimum rawComm is 1, if the node is part of a clique. The maximum rawComm is the degree, if the node is the center of a star. In the extremely rare network where the largest degree nodes are in cliques, the smallest degree nodes are stars and the others are inbetween, the distribution would go from upper left to lower right. However, in our studies we have never experienced such a network where the degreePct and commPct are negatively correlated.

The Movie data set [uci2] contains 6791 Hollywood actors and the movies they starred in. From the data we built a network, shown in Figure 8, of the actors with links between actors who co-starred together in at least one movie. We also assembled a file of students who belonged to the web site FaceBook. Members of the site can list friends of theirs (also on FaceBook). The network shown in Figure 9 was built using 1,030 students from a college in Michigan with links between friends. Looking at the charts for both sets one can see that they do in fact have the kind of shape we expected. Both sets tend to have more loners than other

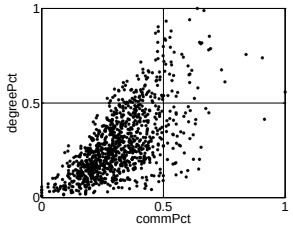

Figure 8: Face Book

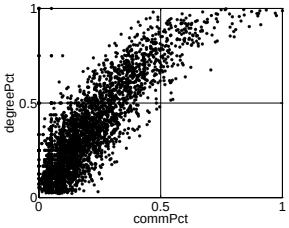

Figure 9: Movies

roles. We would expect this in a scale-free type of network where there are many more low degree nodes and than high.

Inspecting the movie data more closely we selected some actors to represent the different roles. As an ambassador, Burt Lancaster had a career which spanned many decades in which he accumulated many costars but also starred in many diverse films (and so belonged to many communities). A typical Big Fish is Geraldine Chaplin who had a relatively large number of costars but many of the costars were costars of each other - not surprising as nine of her films had the same director. There were no Bridges, strictly speaking, but actors who were close to being Bridges include Kathleen Turner, Jamie Lee Curtis and Martin Landau, all of whom have taken roles with diverse sets of other actors.

6. COMMUNITY METRIC APPLICATIONS

Analysts in many diverse fields currently use existing metrics like degree and betweenness centrality. We anticipate that many of them may find the roles that we describe in section 3 to be a useful addition to their toolbox. For example, in a database of terror suspects, it may be meaningful to know that a particular individual is, say an ambassador. Although we cannot demonstrate the value that the roles may have by themselves we can show how some current techniques can be improved using the metrics we introduced.

6.1 Influence Maximization

As described above the problem of influence maximization is finding the most influential k nodes in a network. This is important in the new field of viral marketing where word-of-mouth advertising can be very effective but discounts or promotions can be costly so marketers need to be judicious about their choice of customers or nodes to activate.

In our experiments we used the Independent Cascade model discussed in the paper by Kempe, et al [5]. Under this model, influence is spread from node to node in discrete steps. A node i that becomes active in step t has one chance to make his inactive neighbors active in step $t+1$. The probability that node i will activate node j in their paper is called $p_{i,j}$ but to avoid confusion with our p value we will call this the edge weight.

We evaluated six algorithms. We compared the total number of nodes that are activated, which is the original goal of this problem. But we also compared how many groups are reached which is our objective in this experiment. The baseline *random* approach selects k nodes randomly. *degree* selects the k nodes with highest degree. The algorithm proposed by Kempe, et al., labeled *greedy*, chooses one new node each iteration, selecting the node that will result in the greatest increase of active nodes according to the Independent Cascade model. The last three methods use the

Table 4: Comparison of Algorithms using Movie Data

		Group Coverage		
algorithm	**nodes**	**Movie**	**Director**	**Genre**
random	11.136	0.7	2.6	79.1
degree	18.996	4.6	9.2	87.3
greedy	22.084	4.1	9.2	87.3
comm	17.578	4.3	10.3	92.9
ambass	20.052	4.8	10.0	92.9
degPct	13.894	1.8	4.1	71.2

metrics we have proposed in this paper. The method *comm* selects the k nodes with the highest rawComm score and *amb* selects the k nodes with the highest sum of commPct and degreePct, while *degPct* selects the k nodes with the highest degreePct.

6.1.1 Influence Maximization using Movie Data

The algorithms were compared using two groups of data, the actors from the movie data set described earlier and some synthetic data sets. There is a link between the actors who co-starred in at least one movie together. Actors who appeared in only one movie were removed. The network had 3,725 nodes and 58,123 links. To find the activated nodes we used the independent cascade model. For the size of the target set (k) we used 10. All of the edge weights were .01. For each method we calculated the activated nodes 500 times and then averaged the results.

We used the data in the original file to form the hidden groups (communities). For the first set of groups we used the movie that the actors appeared in. There are 10,756 movies and actors belong to all of the movies that they star in. The second set was based on the director of the movie of which there are 2,801. An actor belongs to a director's group if they starred in any of that director's movies. The third set is based on genre. Every movie is associated with one of 14 genres. For all three of the group sets an actor can belong to more than one group.

The results for the movie data are summarized in Table 4. The column labeled *nodes* is the average number of nodes activated by the target 10 nodes. The columns under percent of groups indicates how many groups out of the total had at least one node activated. The greedy method, not surprisingly, was able to activate the greatest number of nodes. However, even though *ambass* activated fewer nodes, it was able to reach more groups. *comm* also was able to spread to a large number of groups even though it selected fewer nodes than *degree*, *greedy* or *ambass*. That is not too surprising given that nodes connected to many groups are not necessarily high degree nodes. The *degPct* performed almost as poorly as random.

We also calculated the p and q values (described in section 3.2) for the movie data set. The p values were 1 for all three groups. The q values were .999, .947 and .164 for movie, director and genre groups respectively. This means that for all three groups if there is a link between two actors there is a 100% chance that the two actors will be in the same group. If there is not a link between them then the q tells us that for group one it is nearly certain that they will not be in the same group whereas with group three there is only a 16% chance that they will be in different groups.

6.1.2 Influence Maximization using Synthetic Data

We wanted to see how the algorithms would behave using different network types. So the next set of experiments use synthetic data sets using the methods described in section 4.1. Our approach was to create the synthetic network, run the algorithms and then to assign the nodes to groups for evaluation. Early investigations explored clustering algorithms (single-link, complete-link and group-average) as well as the Normalized Cut spectral graph partitioning algorithm.

These community finding methods all resulted in undesirable p and q values. Single-link had a very high p value but a q near zero. Complete-link and group-average had the opposite problem - high q but a low p. The Normalized Cut had a higher p value but it was still around only 30%. Since our original conjecture is that the links provide evidence of groups we decided to use cliques (maximal complete components) as our grouping algorithm. Intuitively this is appealing because it allows a node to belong to multiple communities. The downside is that because it is NP-complete we needed to use smaller, sparser data sets.

Recall that in the influence maximization experiment using the movie data set an edge weight of .01 was used for computing the number of nodes that become activated. This is the same edge weight that was used in the study by Kempe et al. In their study, as in ours the accurate edge weights are not known so we choose values that make analysis possible. Choosing the edge weight too small means that very few nodes will be activated and choosing an edge weight too large will result in all nodes being activated no matter what algorithm is used.

Since the synthetic sets are more sparse we used edge weights of .1 and .2 for comparison. For each network type, each algorithm selected 10 nodes to activate. Then, using the Independent Cascade model, nodes were activated randomly 1,000 times and the results averaged. The selection process was repeated twenty times for each algorithm and again the results were averaged resulting in a stable distribution.

The results, shown in Table 5 illustrate how conditions affect the different algorithms. First note that the greedy algorithm always is able to activate the highest number of nodes - that is what it is tuned to do. Activating more nodes does not necessarily translate to covering more groups. With an edge weight of .1 greedy covers fewer groups than any of the algorithms except for random and, in the small world and random networks, comm.

With the edge weight of .1, the best performing algorithms are comm and degPct. Looking back at Figures 8 and 9, we can see that with a scale-free network the majority of nodes are loners which indicates many communities that are separated from each other. In such a case an algorithm the focuses on nodes that have a high rawComm would be more likely to spread to more communities. The small world and random networks have many more big fish which indicates communities more likely to overlap each other. This is good for all the algorithms but particularly degPct since it selects nodes with a high relative degree it is more likely to select nodes that are initially not in the same community.

When the edge weights are changed to .2, greedy performs better. It is the best algorithm in the small world and random networks but under scale-free it still is beat by comm and degree. This is because as the edge weights increase

Table 5: Comparison of Algorithms using Synthetic data

	Edge Wgt=.1		Edge Wgt=.2	
algorithm	**nodes**	**% cov.**	**nodes**	**% cov.**
scale-free network				
random	12	24.3	14	29.4
degree	14	51.6	18	55.3
greedy	15	46.7	20	54.8
comm	14	54.2	18	57.3
ambass	14	48.5	17	49.9
degPct	14	46.0	17	47.1
small world network				
random	14	30.0	21	40.4
degree	16	38.9	22	47.6
greedy	17	38.0	26	50.5
comm	15	36.6	21	45.4
ambass	15	38.2	21	46.9
degPct	15	39.7	22	48.9
random network				
random	14	29.7	20	38.1
degree	15	38.4	22	47.5
greedy	18	38.4	26	51.2
comm	15	36.5	21	45.2
ambass	15	37.9	21	47.0
degPct	15	38.9	21	47.6

greedy does a better job of spreading to other nodes and just by sheer numbers is able to cover more groups.

It is also reassuring to note that the results from the movie experiment correlate best with the scale-free synthetic results. Since the degree distribution for the movie data set follows a power law it would be considered a scale free network. So it is not surprising that it would agree with the results from the scale-free synthetic data set.

When considering which algorithm to choose for maximizing community coverage it is important to know the structure of the network, the edge weights and the nature of the communities. In general, if the network is scale-free, using ambass or comm will most likely be result in the largest number of communities covered. Also, although the greedy algorithm always spreads to more nodes than any of the competitors it is a very slow algorithm. If time is limited, using ambass or even degree are reasonable alternatives.

6.2 Classification

As a second example of how the new metrics can be used in existing applications we again use the FaceBook data set but for a different Michigan university. Previous studies have shown that ordinary classification can be improved by using linked data. We will show that by using the rawComm metric we can better employ the linked data to further improve the results.

Chakrabarti, et al.[2] and Lu and Getoor [7] have shown that using linked data can be helpful in some circumstances. In the former paper, the authors were able to show that in some circumstances using the data from neighbors is not helpful but using the class from neighbors can be. A study by Yang, et al.[14], shows that data sets can contain different types of regularities - with some regularities using the class of neighbors is helpful while in others it is not - confirming Chakrabarti's finding.

Users of the FaceBook website can elect to make their personal data visible to other users in the network (usually within a college or university). We were able to collect about 65% of the personal data for the 3,938 students in our data set. The personal data includes gender, birthdate, relationship status (single, etc.), personal relationship interests (friendship, dating, etc.), political view (i.e. conservative), home town, favorite books, favorite movies, leisure interests (skiing, shopping, etc.) and area of academic concentration. To build a set suitable for classification we discretized several of the features. For the features with lists such as favorite books we created five binary (Y/N) features corresponding to the five most popular responses.

We selected, as the class, the person's political view because it is a feature many organizations would be interested in if missing. There are 9 categories including "not available". Using the data from just the webpage itself using a decision tree classifier the error was 73%.

We suggest using our concept of node role to improve the performance of the classifiers that use the class of neighboring nodes. We conjecture that some node types (ambassadors for instance) may not be as influenced by their neighbors while other types (big fish) may be more influenced by them). Before modifying the data for the classifiers we tested our hypothesis that nodes of different roles are influenced by their neighbors differently.

The results are summarized in Table 6. The second column represents the average (across roles) of the percentage of neighbors that have the same class as a node. Since many of the classifiers discussed in the studies above used a majority vote algorithm we also calculated the percentage of nodes that had the same class as the majority of it's neighboring nodes, shown in column 3.

Table 6: Node class versus neighbor's by role

Role	**% same**	**% same as majority**
loners	33.71	30.00
bridges	31.88	27.78
big fish	38.77	50.00
ambassadors	26.60	12.50

Next we modified the data for the classifiers so that for each instance another feature was added that was the class of the majority of its neighbors. Using the neighbors class significantly improved the performance of the classifier as can be seen in Table 7. The numbers in column two represent the number of correctly classified instances out of 2,556 instances for the decision tree classifier.

The first line is the baseline classifier with no neighbor class information. The second line shows the results when we used the neighbor's class. Finally we list the results when we used the neighbors's class only for Loner and Big Fish nodes. While the improvement is not dramatic it is not unexpected given that over 97% of the nodes were Loners. So the data was not changed significantly from the second test to the third. We expect that improvements will be more pronounced in data sets where there are different distributions of roles.

7. CONCLUSIONS AND FUTURE WORK

We have demonstrated in this paper the usefulness of a metric that measures the approximate number of communi-

Table 7: Classifying using role

description	tree
without neighbors' class	700
with neighbors' class	935
selected neighbors' class	943

ties that a node in a network belongs to and the degree of a node relative to it's immediate neighbors. Also introduced is the concept of community-based role which can reveal hidden characteristics of a node.

We have shown how these new metrics can be used to improve the performance of classifiers and to expand the usefulness of algorithms that maximize the spread of influence. It is possible that there exist many more applications. Additionally the assigning of roles to nodes could be useful itself to analysts.

8. REFERENCES

[1] A.-L. Barabási and E. Bonabeau. Scale-free networks. *Scientific American*, 288:50–59, May 2003.

[2] S. Chakrabarti, B. Dom, and P. Indyk. Enhanced hypertext categorization using hyperlinks. *SIGMOD International Conference on Management of Data*, pages 307–318, 1998.

[3] P. Domingos and M. Richardson. Mining the network value of customers. *Conference on Knowledge Discovery in Data*, pages 57–66, 2001.

[4] L. Getoor and C. P. Diehl. Link mining: a survey. *SIGKDD Explorations*, 2005.

[5] D. Kempe, J. Kleinberg, and E. Tardos. Maximizing the spread of influence through a social network. *Conference on Knowledge Discovery in Data*, pages 137–146, 2003.

[6] J. Kleinberg. Sources in a hyperlinked environment. *Journal of the ACM*, 46, 1999.

[7] Q. Lu and L. Getoor. Link-based classification. *In International Conference on Machine Learning*, 2003.

[8] M. Newman and M. Girvan. Finding and evaluating community structure in networks. *Physical Review E*, 69, Feb 2004.

[9] L. Page, S. Brin, R. Motwani, and T. Winograd. Pagerank citation ranking: Bringing order to the web. *Technical report, Stanford University*, 1998.

[10] J. Scripps and P. N. Tan. Clustering in the presence of bridge-nodes. *Proc of SDM'06: SIAM Int'l Conf on Data Mining, Bethesda, MD*, 2006.

[11] J. Shi and J. Malik. Normalized cuts and image segmentation. *Ieee Transactions On Pattern Analysis And Machine Intelligence*, 22(8), August 2000.

[12] S. Wasserman and K. Faust. *Social Network Analysis: Methods and Applications*. Cambridge University Press, Cambridge, UK, 1994.

[13] D. J. Watts and S. H. Strogatz. Collective dynamics of Śsmall-worldŠ networks. *Nature*, pages 440–442, June 1998.

[14] Y. Yang, S. Slattery, and R. Ghani. A study of approaches to hypertext categorization. *Journal of Intelligent Information Systems*, 18, March 2002.