Emergent Personalized Communities in Referral Networks

Pınar Yolum and Munindar P. Singh
Department of Computer Science
North Carolina State University
Raleigh, NC 27695-7535, USA

{pyolum, mpsingh}@csc.ncsu.edu

Abstract

Consider a decentralized agent-based approach for service location. Here agents provide and consume services, and also cooperate with each other by giving referrals to other agents. Based on feedback from their users, the agents judge the quality of the services provided by others. Further, based on judgments of service quality, the agents also judge the quality of the referrals given by others. The agents can thus adaptively select their neighbors in order to improve their local performance. The agents' choices in terms of whom to interact with cause communities to emerge. Accordingly, an agent belongs to a community only if it has been useful to the other members of the community in prior interactions. Hence, the membership in different communities is determined based on relationships among the agents. These communities are personalized, in that their formation depends on choices made by each member to serve the personalized needs of its user.

This paper compares communities of the above kind (personalized, topic-sensitive) with communities as studied in traditional link analysis. We study the correlation between the two kinds of communities as they emerge in referral networks. We also evaluate the two kinds of communities in terms of their effectiveness in locating service providers.

1 Introduction

Many personalization techniques rely on user models to deliver more personalized services. Generally, these techniques work either at the user side or at the server side. Client side techniques use heuristics to capture a user's preferences, and hence promote user's perspective. An example to client side personalization architecture is that of Cingil *et al.*, where the agents generate dynamic profiles of their users [2000]. Conversely, server side techniques take a system perspective in that they use heuristics to aggregate users' interactions with a service. An important example is that of Mobasher *et al.*, where web usage logs are mined at the servers to discover association rules and to cluster profiles of users [2000].

Traditional work on community mining follow the system's perspective. Rather than mining Web logs, the links among many sources are mined, but still the communities are represented centrally. Here, we develop an approach that is closer to approaches that promote users' perspective, where user agents evaluate specific episodes of service delivery. The agents supporting users are aware of other agents and cooperate with them. Consequently, communities emerge. We term these communities personalized communities because they are based on actions and preferences of users.

The study of networked communities is natural. Since communities exist in the physical world, it is to be expected that they will emerge in the virtual world as well. On the Web, communities can help us both identify sites and topics and fine-tune the experience of each user by giving us a basis for making recommendations [Zhong *et al.*, 2002]. Social network analysis and community mining have garnered much research attention lately. In order to understand existing results as well as to evaluate different approaches, it is important to understand communities from a computational standpoint.

Three main definitions of community are commonly used. These are all graph based, but the vertices and edges are interpreted differently.

- *Sociology*. The original definition comes from social network analysis in sociology [Wasserman and Faust, 1994; Scott, 1991]. The idea here is to understand social relationships of various kinds among people and to analyze those relationships to determine the communities in which those people participate and to understand the social behavior of the people. The relationships between people are a given—they are determined by sociologists, e.g., through ethnographic studies. That is, the vertices are people and the edges are observed social relationships (e.g., kinship or friendship) between them.
- *Static link analysis*. Recently, several approaches have been developed to mine communities from Web pages [Kumar *et al.*, 1999b; 1999a; Flake *et al.*, 2002]. These approaches view populations as graphs in which the edges are unlabeled and do not change (within the model). The vertices are Web pages and the links are hyperlinks from one page to another. These links are assumed to be an endorsement from one party to another.

There is no semantics of the links. Communities are defined as patterns of self-similarity as in co-citations. Large corpora of pages can then be mined centrally to determine communities.

- *Referrals and adaptivity*. These approaches consider interactions among agents (or the people they might represent) [Huhns *et al.*, 1987; Kautz *et al.*, 1997; Singh *et al.*, 2001]. The agents maintain models of each other and help each other find agents by giving referrals. The agents potentially learn about each and adaptively decide which other agents they wish to consider their neighbors. Thus a system of interacting agents can be viewed as an evolving social network [Wellman, 2001]. As a graph, its vertices are the agents and edges are the neighborhood relation.

The rest of this paper is organized as follows. Section 2 studies communities in more depth, with an analysis of link-based community mining. Section 3 describes our referrals-based framework. Section 4 explains our approach for mining communities. Section 5 evaluates our approach by comparing it to a related approach. Section 6 discusses the relevant literature and motivates directions for further work.

2 Understanding Communities

From a computational standpoint, it is important to understand the potential applications of communities. We consider two main classes of applications.

- *Endogenous*. The members of a community use the community to find services (including information). That is, the participants use a community somewhat as people might use their social network to decide what movie to watch or what house to buy. Since the boundaries of communities in real-life are amorphous, the participants may not even be aware of which specific community they are benefitting from.
- *Exogenous*. The community structure is used to make recommendations. For example, a recommender system might use some features of a community to which a user belongs to recommend which movie the user might watch. Conversely, the recommendations might be made to the providers of services so they can fine-tune their offerings for a particular community.

Let's now consider how the above classes of research on communities would function in the context of the above kinds of applications of communities.

The sociological work is not directly applicable in Web-based settings, because the underlying social relationships are not explicit. However, if the underlying relationships can be acquired or inferred, it provides a useful intellectual basis for the computational work. Specifically, sociologists have defined various metrics to measure socially relevant properties of graphs, which can be adapted for analysis of computational communities. An important result here, from our perspective, is of the empirically observed strength of weak ties [Granovetter, 1973]. Weak ties are distant social relationships (i.e., acquaintanceship rather than friendship), but prove effective in various purposes of matchmaking or locating information or services—e.g., helping people find jobs.

Link analysis operates on mined Web pages and so has access to millions of pages. Excellent algorithms have been developed, generally based on assuming a "social" relationship among pages that link to the same pages. However, these approaches have a soft underbelly: the lack of semantics. In simple terms, connection between a link on one's page and a social relationship is tenuous at best. For this reason, although graph structures can be extracted, it is not automatically obvious that these structures correspond to communities as we would intuitively consider them. The following are some of the limitations of link analysis as a basis for applying communities:

- Co-citation as a relationship almost seems to be incidental, whereas one would expect that socially related parties to be at least be aware of one another. With co-citation the participants are not aware of each other. Thus the endogenous applications are ruled out. Interestingly, participants cannot readily determine the communities of which they are members. Also, they cannot easily exit a community if they want to, because the only way out is to quit linking to several pages and hope the co-citation relationship is broken.
- Conceptually, communities are discovered in a central manner. This indicates a grave risk of violating the privacy of the potential participants.
- Clearly, mining can work best for only static Web pages, which the participants have made available publicly.

The referrals-based approach is the one we advocate. Like the sociological approach, it considers interactions among the agents participating in a community. The agents help one another and evaluate each other's effectiveness. Good interactions reinforce their social relationships and bring them closer, whereas bad interactions weaken their social relationships. The agents decide with whom to interact. Intuitively, agents will base their decisions on specific feedback or generic policies set by their users, but in terms of its interactions with other agents, each agent is autonomous.

We imagine that the agents are interested in locating suitable service providers. The agents initially request their neighbors for a suitably described service. A queried agent may (1) offer to provide a service in response to the request, (2) give referrals to some of its neighbors, or (3) ignore the request. A requesting agent may follow some of the referrals it receives and ultimately select a service provider. Thus, in essence, an agent explores its social network. Yu and Singh show that an adaptive referrals-based approach is superior for searching a social network (constructed from coauthorship data) [2003].

The referrals approach has some natural advantages:

- Because agents maintain models of others, they are able to annotate their links to other agents in terms of those models. For example, an agent A may believe that B is the best source for information on travel and C is the best source for information on cooking.

- Referrals are generated dynamically. Thus, instead of merely looking at a static Web page, we can model computations wherein (as it were) the page is produced on demand. The responder (acting as the producer of a Web page) can consider its relationship with the requestor in deciding how to respond. Importantly, the referrals approach, because it involves requests among the participants, can apply on the so-called Deep Web, whereas a conventional mining approach would apply only to the static Web.

No formal community needs to be identified for an agent to function correctly. Personalized communities emerge around each agent and each agent automatically exploits them and evolves them as it goes about its business. No central authority need know what the communities are. However, in order to perform our analysis, we mine the communities. Doing so enables us to compare our approach to the link analysis approach.

Our first comparison is qualitative and considers the kinds of structures identified as communities by link analysis. Intuitively, link analysis definitions prove to be neither necessary nor sufficient to describe real communities.

2.1 Conceptual Analysis

Communities there are defined in terms of related sets of *fans*, which ideally point at lots of centers, and *centers*, which are ideally pointed to by lots of fans. Kumar *et al.* propose that any community structure should contain a bipartite core where the fans and centers constitute the independent sets. If all n fans point to a set of m authoritative pages, then they are likely to share a common topic and therefore be a community. Especially in the case of high n and m, the likelihood of being a community is assumed to be higher. In addition, all other nodes that are pointed by the fans, and all the nodes that point to at least two centers are added. In their experiments, Kumar *et al.* use (3,3) bipartite cores. For example, the graph in Figure 1 could denote a community where the solid lines show the links that make up the bipartite core, and the dotted lines are added by expanding the core. Even though there is a link between nodes 6 and 9, 9 is not added to the community. We refer to these as bipartite communities.

The underlying assumption in these approaches is that fans all share one topic. Thus, if many of them point to the same centers, then they must be sharing the same topic. Assuming a single topic for a web page can be realistic, but agents can be interested in different domains as well as providing services in different domains.

We consider the relation between communities and bipartite cores from two directions. The first direction is to see if a bipartite core always denotes a community. Consider three consumers interested in the same three domains. Each one of them chooses the same three service providers, each of whom provides a service in one domain. They form a bipartite core since all the consumers point at all these three providers. Obviously though, this is not a community. The three service providers are not even providing the same service.

The other direction is to look at if every community should have a bipartite core. There might very well be graphs that

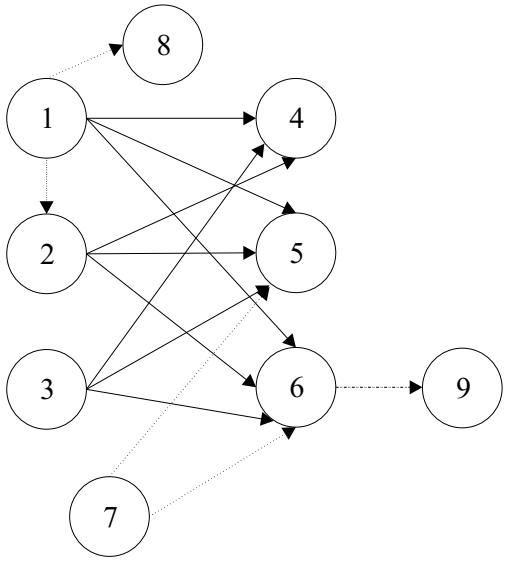

Figure 1: A community as defined by Kumar *et al.*

do not contain a (3,3) bipartite core but instead are well-connected like the one in Figure 2. Could this still be a community?

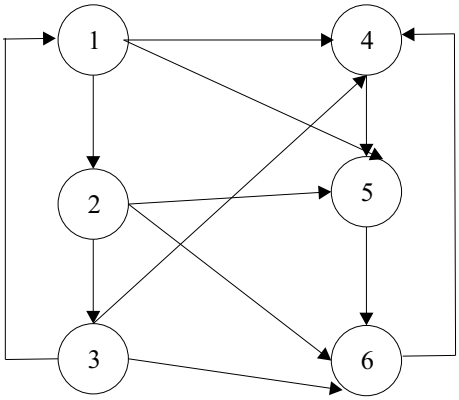

Figure 2: A possible community without a bipartite core

The structures may be interpreted differently in different settings. For example, an independent set of service providers is common in commerce: the service providers are not looking for services themselves, so they do not point to each other. In knowledge management, on the other hand, every agent is potentially looking for services. Having an independent set of agents can have several implications, such as different evaluation of services, being unaware of each other, and so on. Hence, the structures alone may not be sufficient to accurately represent communities.

2.2 Quantitative Comparison

In addition to the link structure, the properties of the links among the agents are interesting. Especially important among these are those properties that can be useful in determining the strength of the links. Strength of the links can

augment the structural mining of communities. For instance, in some cases, to conclude that the agent is part of a community, it might be enough to show that it has one strong link to a member of a community, whereas if the agent has weaker links to the community members, more links might be required.

Comparing referrals approaches with link analysis in quantitative terms is potentially tricky, because link analysis is applied on actual Web pages, whereas referrals apply between agents providing and seeking services. There is no widespread practical deployment of such service location schemes. However, a comparison is possible when we consider how links are created. Conventional models of the Web function at a gross level, statistically applying some rule such as preferential attachment [Barabási *et al.*, 2000]. But, in fact, the creation of links on the Web is based on micro evaluations and decisions by independent players. This process of neighbor selection is mimicked well by adaptive referrals.

This paper accordingly proceeds with the following methodology. We simulate a referral network in which the agents evolve a social network. From this network, we can infer link analysis communities as well as referrals based communities. We must introduce our technical framework before giving the details, but our main results are as follows. One, the two kinds of communities are uncorrelated. Two, referral-based communities yield greater quality both for endogenous and exogenous applications.

3 Technical Framework

The agents act in accordance with the following abstract protocol. An agent begins to look for a trustworthy provider for a specified service. The agent queries some other agents from among its *neighbors*. A queried agent may offer to provide the specified service or may give referrals to other agents. The querying agent may accept a service offer, if any, and may pursue referrals, if any. Each agent maintains models of its acquaintances, which describe their *expertise* (i.e., quality of the services they provide) and *sociability* (i.e., quality of the referrals they provide). Both of these elements are learned based on service ratings from its principal. Using these models, an agent applies its *neighbor selection policy* to decide on which of its acquaintances to keep as neighbors. Key factors include the quality of the service received from a given provider, and the resulting value that can be placed on a series of referrals that led to that provider. In other words, the referring agents are rated as well. An agent's own requests go to some of its neighbors. Likewise, an agent's referrals in response to requests by others are also given to some of its neighbors, if any match. This, in a nutshell, is our basic social mechanism.

The neighborhood relations among the agents induce the structure of the given society. In general, as described above, the structure is adapted through the decisions of the different agents.

3.1 Applicable Domains

The above framework enables us to represent different application domains naturally. Two important domains are commerce and knowledge management, which have differ in their notions of service and how the participants interact.

In a typical commerce setting, the service providers are distinct from the service consumers. The service consumers lack the expertise in the services that they consume and their expertise doesn't get any better over time. However, the consumers are able to judge the quality of the services provided by others. For example, you might be a consumer for auto-repair services and never learn enough to provide such a service yourself, yet you would be competent to judge if an auto mechanic did his job well. Similarly, the consumers can generate difficult queries without having high expertise. For example, a consumer can request a complicated auto-repair service without having knowledge of the domain.

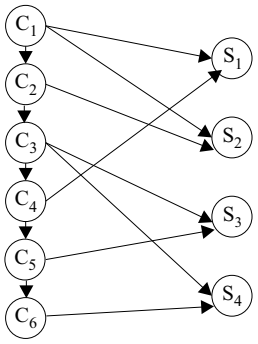

Figure 3: A schematic configuration for e-commerce

By contrast, in knowledge management, the idea of "consuming" knowledge services would correspond to acquiring expertise in a given domain. A consumer might lack the ability to evaluate the knowledge provided by someone who has greater expertise. However, agents would improve their knowledge by asking questions; thus their expertise would increase over time. Following the same intuition, the questions an agent generates would also depend on its expertise to ensure that the agent doesn't ask a question whose answer it already knows.

Figure 3 is an example configuration of service consumers and providers that corresponds to a commerce setting. The nodes labeled C denote consumers and the nodes labeled S denote service providers. Consumers are connected to each other as well as to the service providers. These links are essentially paths that lead to service providers with different expertise. In this model, the service providers are dead ends: they don't have outgoing edges, because they don't initiate queries or give referrals. Thus, their sociability stays low. Their true and modeled expertise may of course be high.

3.2 Evaluation Architecture

We have implemented a distributed platform using which adaptive referral systems for different applications can be built. However, we investigate the properties of interest over a simulation, which gives us the necessary controls to adjust various policies and parameters. The simulation involves n agents, a large fraction of whom are service consumers looking for providers. Consumers have high *interest* in getting

different types of services, but they have low expertise, since they don't offer services themselves. Providers have high expertise but low sociability. The interests and expertise of the agents as well as their modeled sociability are represented as term vectors from the vector space model (VSM) [Salton and McGill, 1983], each term corresponding to a different domain. The simulation uses these to generate queries and answers for the various agents.

Algorithm 1 Ask-Query()

```
1: Generate query
2: Send query to matching neighbors
3: while (!timeout) do
4:    Receive message
5:    if (message.type == referral) then
6:       Send query to referred agent
7:    else
8:       Add answer to answerset
9:    end if
10: end while
11: for i = 1 to |answerset| do
12:    Evaluate answer(i)
13:    Update agent models (expertise & sociability)
14: end for
```

Each agent is initialized with the same model for each neighbor; this initial model encourages the agents to both query and generate referrals to their neighbors. An agent that is generating a query follows Algorithm 1. An agent generates a query by slightly perturbing its interest vector, which denotes that the agent asks a question similar to its interests (line 1). Next, the agent sends the query to a subset of its neighbors (line 2). The main factor here is to determine which of its neighbors would be likely to answer the query. We usually determine this through the capability metric.

An agent that receives a query acts in accordance with Algorithm 2. An agent answers a question if its expertise matches a question. If the expertise matches the question, then the answer is the perturbed expertise vector of the agent. When an agent does not answer a question, it uses its *referral policy* to choose some of its neighbors to refer.

Algorithm 2 Answer-Query()

```
1: if hasEnoughExpertise then
2:    Generate answer
3: else
4:    Refer neighbors
5: end if
```

Back in Algorithm 1, if an agent receives a referral to another agent, it sends its query to the referred agent (line 6). After an agent receives an answer, it evaluates the answer by computing how much the answer matches the query (line 12). Thus, implicitly, the agents with high expertise end up giving the correct answers. After the answers are evaluated, the agent uses its *learning policy* to update the models of its neighbors (line 13). In the default learning policy, when a good answer comes in, the modeled expertise of the answering agent and the sociability of the agents that helped locate the answerer (through referrals) are increased. Similarly, when a bad answer comes in, these values are decreased. At certain intervals during the simulation, each agent has a chance to choose new neighbors from among its acquaintances. Usually the number of neighbors is limited, so if an agent adds some neighbors it might have to drop some neighbors as well.

4 Community Mining

Previous approaches view communities as crisp structures in that an agent is either a member of a community or not. On the other hand, in general there are no clear-cut boundaries for communities. A community may have many members who differ in their level of belonging to the community. Accordingly, our approach is based on ranking members of a community based on their level of membership. An agent may belong to several communities in varying levels.

4.1 Methodology

We consider mining communities of service consumers for different domains. As an example consider travel domain. There are several travel agents represented as service providers. Some service consumers are interested in finding travel agents and query other service consumers to locate the providers. The service consumers who help find the travel agents are found to be sociable by the travelers, since the sociable agents' referrals help in locating the providers. Hence, a consumer is part of the travelers if it has found to be sociable by other agents. A consumer belongs more to a community if more consumers find him to be sociable and these consumers themselves are part of the community. Members of the community decide who should be in the community. This recursive definition is inspired by the PageRank algorithm.

PageRank. PageRank is a metric used by Google to rank Web pages that are returned for a query [Brin and Page, 1998]. The PageRank of a Web page measures its authoritativeness. Informally, a Web page has a high PageRank only if it is pointed to by Web pages with high PageRanks, i.e., if other authoritative pages view this page as authoritative. We use the same metric to measure the authoritativeness of agents. The PageRank of an agent is calculated using Equation 1, where $P(i)$ denotes the PageRank of agent i, I_i denotes agents that have i as a neighbor, and N_j denotes the agents that are neighbors of j. The PageRanks are normalized using a constant d, where d is taken to be 0.85 as in the original paper [Brin and Page, 1998].

$$P(i) = d\sum_{j \in I_i} \frac{P(j)}{|N_j|} + (1 - d) \quad (1)$$

As mentioned above, the neighborhood relations among the agents induce a directed graph, where each node denotes an agent. An edge (u, v) exists if u values v's expertise, sociability, or both. This valued expertise or sociability may be in one or more domains.

Referral Communities. The PageRank calculations for the Web are performed on a directed unlabeled graph. Here, we

build on this idea to find communities. First, the graph structure is enhanced by adding labels to the edges of the graph, where the label on an edge (u, v) denotes v's sociability from u's point of view (in our notation, this is $\sigma_{u,v}$) for one domain. u may model v's sociability for different domains. In other words, u might find v sociable for one domain, but not sociable for many other domains. Second, the sociability ranks for each agent is calculated per domain as given in Equation 2. Below, $S(i)$ denotes the sociability rank of agent i, I_i denotes agents that have i as a neighbor, N_j denotes neighbors of j, $\sigma_{j,i}$ denotes the sociability of i for j.

$$S(i) = d\sum_{j \in I_i} (S(j) * \frac{\sigma_{j,i}}{\sum_{k \in N_j} \sigma_{j,k}}) + (1 - d) \qquad (2)$$

In PageRank calculations, at each iteration, each node distributes its PageRank to its neighbors equally. Here, on the other hand each node distributes its sociability rank based on the sociability weights on the edges.

Example 1 Consider an agent j with neighbors i, k, and l such that $\sigma_{j,i} = 0.8$, $\sigma_{j,k} = 0.2$, and $\sigma_{j,l} = 0.2$. j will contribute $\frac{0.8}{0.8+0.2+0.2}$ to i's sociability rank. ▮

This definition of communities captures two important notions. One, members of the communities decide on the other members. Two, the members are chosen based on how helpful they have been to others. This implies that an agent may belong to a community more than a second agent even though both agents have the same neighbors.

4.2 Evaluation

We evaluate our approach of finding communities with respect to bipartite communities.

Capability. The *capability* of an agent for another agent measures how similar and how strong the expertise of the agent is for the second agents' interests [Singh *et al.*, 2001; Yolum and Singh, 2003]. Capability resembles cosine similarity but also takes into account the magnitude of the expertise vector. What this means is that expertise vectors with greater magnitude turn out to be more capable for the interest vector. In (3), I refers to an interest vector, E refers to an expertise vector and n refers to their length.

$$I \otimes E = \frac{\sum_{t=1}^{n} (i_t e_t)}{\sqrt{n \sum_{t=1}^{n} {i_t}^2}} \qquad (3)$$

Utility. The *utility* of an agent denotes how easily it can access the information it needs. The utility of an agent depends on the utility of its outgoing edges and defined as the sum of the utilities of its out-edges. Equation 4 is used to calculate the utility of an edge. An edge has a high utility, if (1) the outgoing edges lead to service providers whose expertise match the agent's interests or (2) if the edges lead to other agents with high values. The first part of the equation is straightforward as defined by the capability metric. For the second part, the agent j can lead to high valued agents by giving a referral to a agent k among its neighbors. Intuitively, j will give a referral to the agent that provides most utility to itself; i.e., the agent k maximizes the utility of (j, k). The usefulness of k for i is then calculated again by $\mu_{i,k}$.

$$\mu_{i,j} = \delta(I_i \otimes E_j) + (1 - \delta)\mu_{i,k} \qquad (4)$$

where

$$k = \arg\max(\mu_{j,k}), \forall k \in N_j \qquad (5)$$

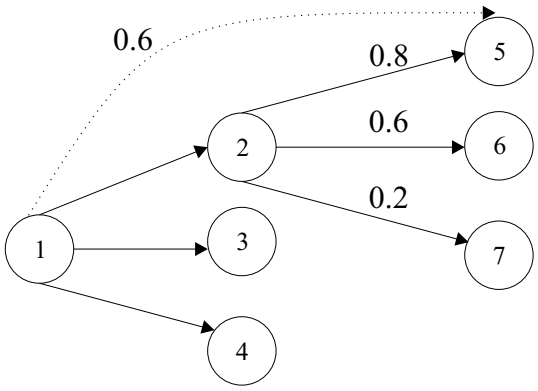

Figure 4: An example for utility computation

Example 2 Consider the example in Figure 4 where the labels denote the utility of the edges and solid lines denote neighborhood relations. To calculate the utility of edge $(1, 2)$ $(\mu_{1,2})$, first $I_1 \otimes E_2$ is calculated. Then, among agent 2's neighbors, agent 5 yields highest utility since the edge $(2, 5)$ has the highest utility. Hence, if agent 2 cannot answer agent 1's query, it would give a referral to agent 5. The contribution of agent 5 to agent 1 is then $\mu_(1, 5)$. Hence, $\mu_(1, 2) = \delta(I_1 \otimes E_2) + (1 - \delta)\mu_{1,5}$. ▮

5 Results

We evaluate our approach by comparing it to bipartite communities. To generate bipartite communities, we find bipartite cores of size (6,3). We expand each core by adding all the nodes pointed by the fans and all the nodes that point to at least two centers. Then, we run HITS algorithm [Kleinberg, 1999] to find the authorities and hubs. The nodes of the community are then ranked based on the hubness values, since the communities are targeted for locating services.

5.1 Correlation

First, we calculate the correlation between communities found by both approaches using Spearman correlation, given in Equation 6.

$$\rho(C, D) = 1 - \frac{6 \sum_{i=1}^{n} (C_i - D_i)^2}{n(n^2 - 1)} \qquad (6)$$

A correlation value of 1 shows that the members of the communities are ranked the same in both approaches, whereas a correlation value of -1 shows that the members of the two communities are ranked in reverse order. Correlation values around 0 denote that the rankings are not correlated.

Below, C and D denote two communities, C_i and D_i denote the rank of agent i in communities C and D, respectively, and n denotes the size of the communities. When comparing the two communities, the community size, n is taken to be the size of the bipartite community found. The top n agents from our ranking is then taken for comparison.

We choose 10 communities for comparison. The choice for the communities is arbitrary, except that the chosen communities vary in their size, where the smallest has community has 32 members and the largest community had 238. The average correlation among the communities is -0.65, with the correlation values varying from -0.3 to -0.9. The fact that there is no positive correlation between the communities means that the rankings of the two communities do not agree. Based on preliminary studies on the distribution of the correlations, we conjecture that as the size of the communities increase, the ranking of the communities become less correlated; i.e., the absolute value of $\rho(C, D)$ approaches 0.

5.2 Utility

The community of service consumers for a service should be able to locate the service providers easily. The utility metric (Equation 4) captures this intuition.

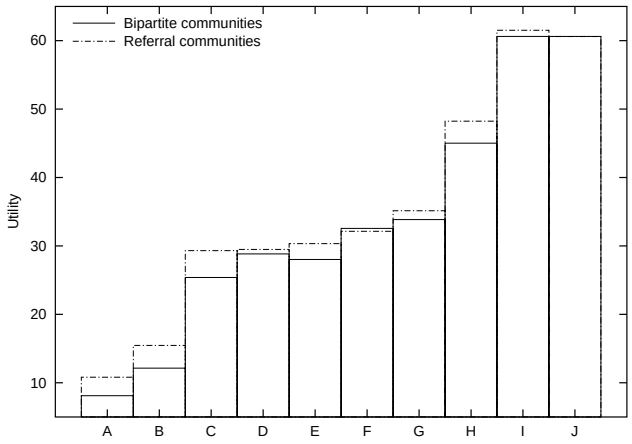

Figure 5: Utility comparison for some communities

We compare referral and bipartite communities in their total utility. Figure 5 gives a histogram of this comparison. The solid lines denote bipartite communities and the dashed lines denote referral communities. Nine of the referral communities outperform bipartite communities in their utility. Only referral community 6 receives a slightly worse utility than the bipartite community.

5.3 One-size doesn't fit all

We study the authorities of a community in terms of how well they serve the query needs of the community members. On one side, for a bipartite community, we rank the members based on their authority from the HITS algorithm. From the community, we make five agents generate queries and ask them to the top four authorities. On the other side, we make

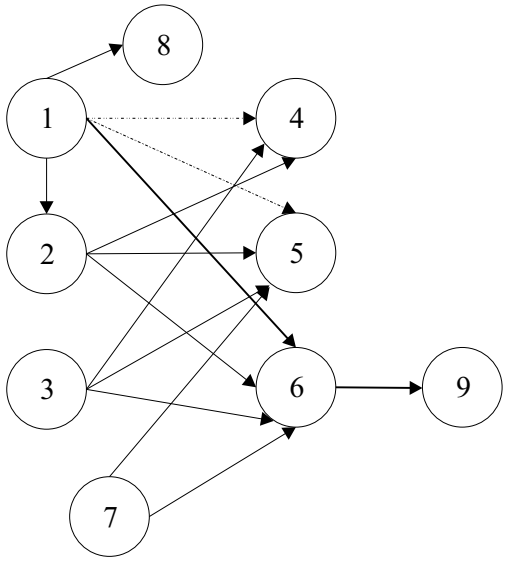

Figure 6: Consider the community of Figure 1. After running HITS, agents 4 and 5 are found to be authorities. In the case for bipartite communities, agent 1 generates a query and asks it to 4 and 5. For the referral communities, it asks the query to its choice of neighbors, in this case agent 6 who gives a referral to agent 9 (bold lines).

the agents look for answers to the same queries through a referral process as shows in Figure 6.

Figure 7 plots the number of good answers for each agent. Four of the five agents get more good answers following referrals, rather than posing their query to the authorities. The last agent get equal number of answers with both approaches. The striking result here is that the authorities that are optimized for everybody's needs are not always effective for individual needs. On the other hand, when agents follow referrals from their personal social network, they can find more useful answers.

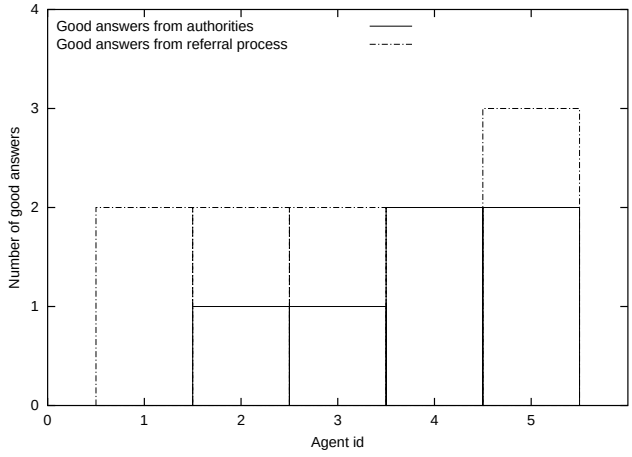

Figure 7: Comparison of good answers

6 Discussion

The literature on social networks [Wasserman and Faust, 1994; Scott, 1991] views communities as cohesive subgroups. It takes three directions to find communities. The first set of approaches exploit the reachability of subgroup members. Although there are subtle differences between these approaches, the main idea is that members of the subgroup should be able to reach each other in as few steps as possible. The second set of approaches exploit the frequency of ties among members, such that removing any member of the subgroup should affect the connectedness of the subgroup as little as possible. The third set of approaches focus on the frequency of ties among the subgroup members versus the frequency of the ties to nonmembers.

The intuition for this last set of approaches underlies Flake *et al.*'s definition of a web community. Flake *et al.* [2002] define a web community as a collection where each page has more links to the members of the community than to pages outside. They model the graph as a maximum flow problem, as follows. First, a set of seed pages are assumed to be in the community. These pages form the source of the maximum flow. A set of portal sites (such as Yahoo!, because of their high indegree) are then connected into a virtual sink node. The minimum cut of the graph separates a community from the rest of the graph and the component that contains the source nodes make up the community. There are two major drawbacks to defining communities this way. First, by this definition, each page can only belong to one community. Second, even if a page has only one link to a community and no links to pages outside the community, it is considered part of the community.

Our approach provides opportunities for further research. In our future work, we want to consider richer sociological ideas such as the ones in social networks as well as compare our approach to other definitions of Web community.

7 Acknowledgments

This research was supported by the National Science Foundation under grant ITR-0081742. We thank the anonymous reviewers for helpful comments.

References

[Barabási *et al.*, 2000] Albert-László Barabási, Réka Albert, and Hawoong Jeong. Scale-free characteristics of random networks: The topology of the World Wide Web. *Physica A*, 281:69–77, 2000.

[Brin and Page, 1998] Sergey Brin and Lawrence Page. The anatomy of a large-scale hypertextual Web search engine. *Computer Networks and ISDN Systems*, 30(1–7):107–117, 1998.

[Cingil *et al.*, 2000] Ibrahim Cingil, Asuman Dogac, and Ayca Azgin. A broader approach to personalization. *Communications of the ACM*, 43(8):136–141, August 2000.

[Flake *et al.*, 2002] Gary William Flake, Steve Lawrence, C. Lee Giles, and Frans M. Coetzee. Self-organization and identification of Web communities. *IEEE Computer*, 35(3):66–70, March 2002.

[Granovetter, 1973] Mark Granovetter. The strength of weak ties. *American Journal of Sociology*, 78:1360–1380, 1973.

[Huhns *et al.*, 1987] Michael N. Huhns, Uttam Mukhopadhyay, Larry M. Stephens, and Ronald D. Bonnell. DAI for document retrieval: The MINDS project. In Michael N. Huhns, editor, *Distributed Artificial Intelligence*, pages 249–283. Pitman/Morgan Kaufmann, London, 1987.

[Kautz *et al.*, 1997] Henry Kautz, Bart Selman, and Mehul Shah. The hidden Web. *AI Magazine*, 18(2):27–36, 1997.

[Kleinberg, 1999] Jon Kleinberg. Authoritative sources in a hyperlinked environment. *Journal of the ACM*, 46(5):604–632, 1999.

[Kumar *et al.*, 1999a] Ravi Kumar, Prabhakar Raghavan, Sridhar Rajagopalan, and Andrew Tomkins. Extracting large-scale knowledge bases from the Web. In *Proceedings of the 25th Very Large Databases Conference*, pages 639–650, 1999.

[Kumar *et al.*, 1999b] Ravi Kumar, Prabhakar Raghavan, Sridhar Rajagopalan, and Andrew Tomkins. Trawling the Web for emerging cyber-communities. In *Proceedings of the World Wide Web Conference*, pages 1481–1493, 1999.

[Mobasher *et al.*, 2000] Bamshad Mobasher, Robert Cooley, and Jaideep Srivastava. Automatic personalization based on Web usage mining. *Communications of the ACM*, 43(8):142–151, August 2000.

[Salton and McGill, 1983] Gerard Salton and Michael J. McGill. *An Introduction to Modern Information Retrieval*. McGraw-Hill, New York, 1983.

[Scott, 1991] John Scott. *Social Network Analysis: A Handbook*. Sage Publications, London, 1991.

[Singh *et al.*, 2001] Munindar P. Singh, Bin Yu, and Mahadevan Venkatraman. Community-based service location. *Communications of the ACM*, 44(4):49–54, April 2001.

[Wasserman and Faust, 1994] Stanley Wasserman and Katherine Faust. *Social Network Analysis*. Cambridge University Press, New York, 1994.

[Wellman, 2001] Barry Wellman. Computer networks as social networks. *Science*, 293:2031–2034, September 2001.

[Yolum and Singh, 2003] Pınar Yolum and Munindar P. Singh. Emergent properties of referral systems. In *Proceedings of the 2nd International Joint Conference on Autonomous Agents and MultiAgent Systems (AAMAS)*. ACM Press, July 2003. To appear.

[Yu and Singh, 2003] Bin Yu and Munindar P. Singh. Searching social networks. In *Proceedings of the 2nd International Joint Conference on Autonomous Agents and MultiAgent Systems (AAMAS)*. ACM Press, July 2003. To appear.

[Zhong *et al.*, 2002] Ning Zhong, Jiming Liu, and Yi Yu Yao. In search of the wisdom Web. *IEEE Computer*, 35(11):27–31, November 2002.