Dynamic 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, where agents provide and consume services, and also cooperate with each other by giving referrals to other agents. That is, the agents form a *referral network*. Based on feedback from their users, the agents judge the quality of the services provided by others. Further, based on the 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 choices by the agents cause communities to emerge. According to our definition, an agent belongs to a community only if it has been useful to the other members of the community in prior interactions regarding a particular topic. Hence, the membership in different communities is determined based on relationships among the agents. This paper compares topic-sensitive communities of the above kind with communities as studied in traditional link analysis. It studies the correlation between the two kinds of communities as they emerge in referral networks and evaluates the two kinds of communities in terms of their effectiveness in locating service providers.

1 Introduction

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 identify interesting and important sites and topics. They can help us fine-tune the experience of each user by giving us a basis for making recommendations [Zhong et al., 2002]. For these reasons, 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 in each.

- *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 they participate. 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 the social relationships (e.g., kinship or friendship) observed between them.

- *Static link analysis*. Recently, several approaches have been developed to mine communities from Web pages [Kumar et al., 1999; 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 edges are hyperlinks from one page to another. A hyperlink from page A to page B is assumed to be an endorsement from the author of page A to page B. There is no semantics of the links. Communities are defined as patterns of self-similarity as in co-citations. Large corpora of pages can thus 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 other useful 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 represent the neighborhood relation. Communities can then be defined from these social networks.

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 4.2 explains our approach for mining communities and introduces metrics for evaluating effectiveness of communities. Section 4 evaluates our approach by comparing it to a related approach. Section 5 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. There are two main classes of applications.

- *Endogenous*. The members of a community use the community to find services (including information services). That is, the participants use a community somewhat as people might use their *personal network* to decide what movie to watch or what house to buy. Their personal network involves a part of the social network that is closely related to them. Since the boundaries of communities in real life are amorphous, the participants may not be, do not need to be, and usually are not aware of the specific community from which they happen to benefit.

- *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 a movie for the user to 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.

Link-based definitions of communities use a form of co-citation. Perhaps the best known of the existing approaches is that of Kumar *et al.* [1999]. This is reminiscent of hubs and authorities in Hyperlink Induced Topic Search (HITS), a well-known Web page ranking technique [Kleinberg, 1999]. The HITS algorithm assigns an authority and a hubness value to each node in the graph. A node has high authoritativeness if nodes with high hubness values point to it. Similarly, a node has a high hubness value if it points to authoritative nodes. Reflecting a similar intuition, Kumar *et al.* define communities 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 two independent sets. If all n fans point to a set of m centers, 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, in the graph in Figure 1, the nodes 1 to 8 could denote a community, where the nodes 1 to 3 are the fans, and the nodes 4 to 6 are the centers. These nodes constitute the bipartite core (shown in solid lines). The node 7 is added because it points to two centers (5 and 6) and the node 8 is added because it is pointed to by a fan (1). These expansions are shown in dotted lines. Even though there is a link between nodes 6 and 9, node 9 is not added to the community. We refer to these communities as bipartite communities.

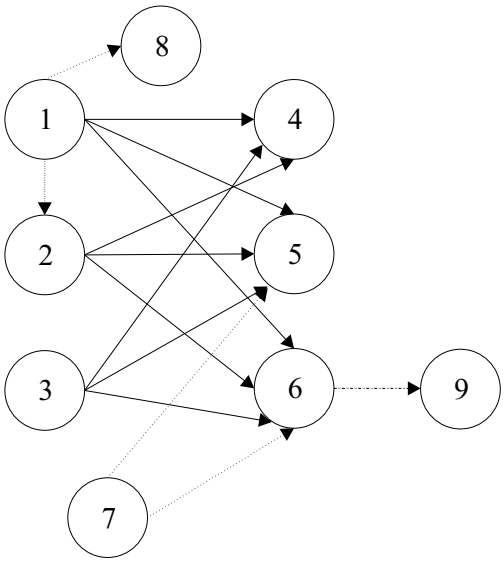

Figure 1: Nodes 1 to 8 form a bipartite community

The following are some of the limitations of link analysis as a basis for applying communities:

- The semantics of the links between the nodes is not defined. Consider three consumers (nodes 1 to 3) interested in distinct three domains. Each one of them chooses the same three service providers

(nodes 4 to 6), each of whom provides a service in one domain. They form a bipartite core since all the consumers point at all these three providers. It is not clear why this structure would denote a community, when the three service providers are not even providing the same service.

- Co-citation as a relationship almost seems to be incidental among the parties so related, whereas one would expect that socially related parties to potentially traverse the network to each other. With co-citation the participants are not aware of each other. For example, nodes 2 and 3 may not know they are in the same community. Thus the endogenous applications are ruled out.

- The structures may be interpreted differently in different settings. For example, consider the two application domains discussed in Section 3.1. 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 (i.e., knowledge in this case). 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.

- 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. Each link is interpreted as an endorsement. However, this approach fails to apply when participants decide not to reveal their endorsements publicly. Conversely, the beneficiary of any recommendation may prefer these from the parties of which it knows.

Because of the above limitations, we advocate the referrals-based approach. Like the sociological approach, the referrals-based approach 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 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 [Raghavan and Garcia-Molina, 2001; Singh, 2002], whereas a conventional mining approach would apply only to the static Web.

Intuitively, link analysis definitions prove to be neither necessary nor sufficient to describe real communities. No formal community needs to be identified for an agent to function correctly. Communities emerge around each agent. 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.

Comparing referrals approaches with link analysis in quantitative terms is potentially tricky, because link analysis is applied on static 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 mutually uncorrelated. Two, referral-based communities yield greater utility both for endogenous and exogenous applications.

3 Technical Framework

The agents in a referral network 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 (i.e., agents that it interacts with), 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 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 of them match a given request. The neighborhood relation among the agents induces 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.

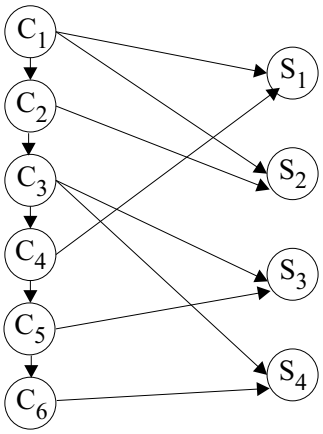

Figure 2: A schematic configuration for e-commerce

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 2 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 400 agents: 5% of them are service providers, and the remaining agents 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 multidimensional and are represented as term vectors in the vector space model (VSM), well-known from Information Retrieval [Salton and McGill, 1983]. Each term corresponds to a different domain. The simulation uses these vector to generate queries and answers for the various agents.

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

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
```

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.

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 decide which of its neighbors, if any, to include in referrals to the requesting agent.

Algorithm 2 Answer-Query()

```
1: if hasEnoughExpertise then
2:    Generate answer
3: else
4:    Generate referrals to 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 updates the models of its neighbors (line 13). 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.

Example 1 Figure 3 shows an example network, where the nodes denote agents. Agent 1's neighbors are agents 2 and 3, agent 2's neighbors are agents 4 and 5, and agent 3's neighbors are agents 5 and 6. Agent 1 poses its query to its neighbors, agents 2 and 3. Agent 2 provides an answer, while agent 3 gives a referral to one of its own neighbors, agent 5. Agent 1 then sends its query to agent 5. Even though Figure 3 shows only 6 agents, our actual experiments contain 400 agents. ∎

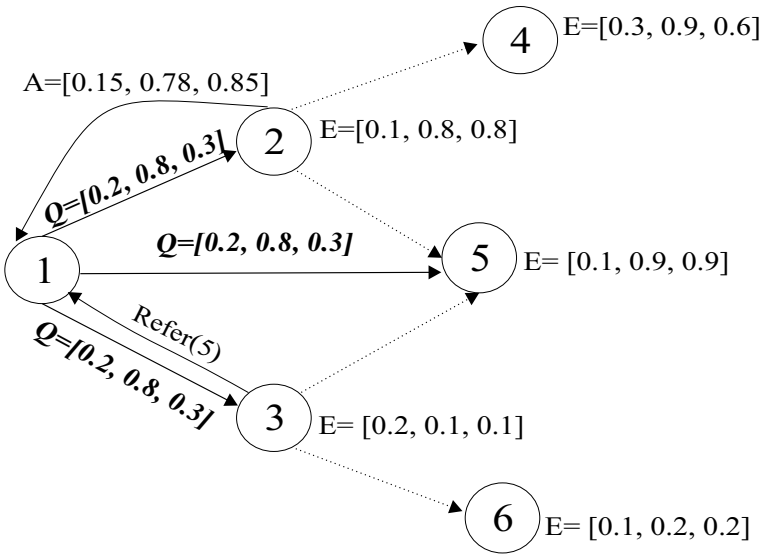

Figure 3: An example search through referrals

4 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 the HITS algorithm (mentioned in Section 2) to find the authorities and hubs. A node has high authoritativeness if nodes with high hubness values point to it. Similarly, a node has a high hubness value if it points to authoritative nodes.

4.1 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. 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 pose them to the top four authorities in the given community. For a referrals-based community, we make the agents look for answers to the same queries through a referral process, as shown in Figure 4.

Figure 5 plots the number of good answers for each agent. Four of the five agents get more good answers by following referrals, than by posing their query to the authorities. The last agent gets an equal number of answers with both approaches. The striking, but perhaps obvious, result here is that the authorities identified by link-based methods are not always effective for individual needs. That is, authorities chosen by others may not serve the needs of every agent. Further, when agents follow referrals from their personal network, they can find more useful answers than otherwise. Agents choose their neighbors based on their previous interactions. A neighbor of an agent has either given good answers or useful referrals. Hence, by following referrals from its neighbors, each agent finds more answers that suit its needs.

This is an especially important result for endogenous settings, where the community is used by the members (as opposed to by external entities) to find desired services. Naturally, how easily agents find useful providers is affected by the agents' own policies [Yolum and Singh, 2003].

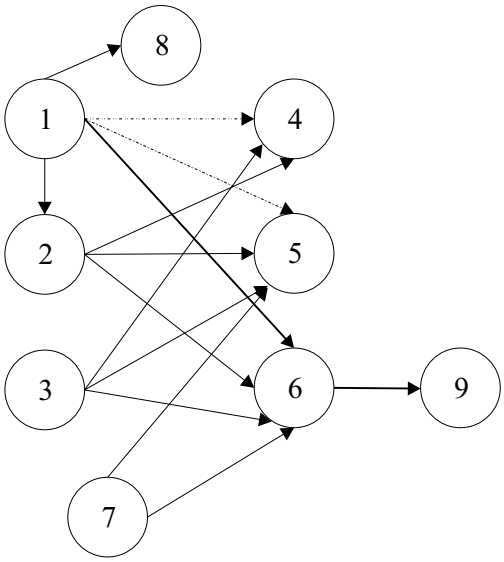

Figure 4: Consider the community of Figure 1. After running HITS, agents 4 and 5 are found to be authorities. In the case for bipartite communities, agent 1 generates a query and asks it to 4 and 5. For the referral communities, it asks the query to its choice of neighbors; in this case only to agent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

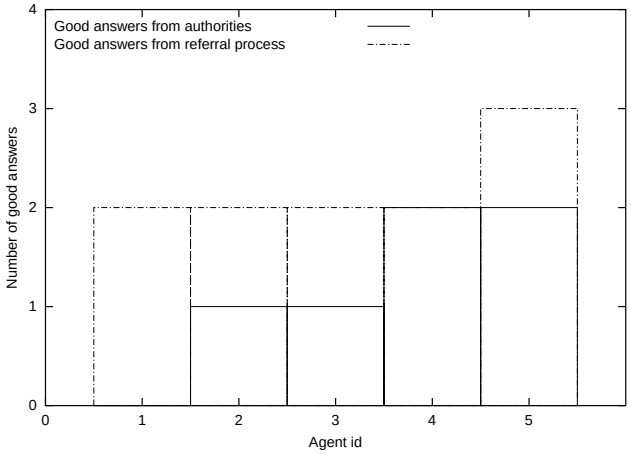

Figure 5: Comparison of good answers for 5 agents

4.2 Community Mining

For endogenous applications, communities can be used without being mined. In general, as we explained above, considerations of privacy preclude the mining of communities by a central entity. That is, most users will not be willing to expose their links publicly to a search engine to reveal to others; instead, users will often just give referrals to suitably trusted parties.

However, there are two reasons why communities may be mined. First, communities would need to be mined for purely scientific purposes, specifically, to compare our approach to bipartite communities. Second, in some environments (e.g., in trusted enterprise settings), it may be possible to mine communities for certain exogenous applications.

Based on the above motivations, we now present an approach for mining referral communities in a manner that facilitates comparison with bipartite communities. In mining communities, especially referral communities, two properties are worth considering.

- Communities may not have clear-cut boundaries. Previous approaches view communities as crisp structures in that an agent is either a member of a community or not. On the other hand, 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.

- Strength of the links matter. 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.

We consider mining communities of service consumers for different domains. As an example consider the 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. Since sociable agents are more influential in locating the service providers, we use sociability of the agents to rank their involvement in the community. A consumer belongs more to a community if more consumers find it to be sociable. Note that sociability is subjective. For instance, agent A may view agent B as sociable, whereas a third agent may not. When this is the case, the agent who is part of the community has a bigger say. In other words, if A is part of the community, it can judge B's contribution better than someone outside the community, or someone less involved in the community. In this regard, 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].

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

Referral Communities. The PageRank calculations for the Web are performed on a directed unlabeled graph. Here, we build on this idea to mine communities. As mentioned above, the neighborhood relation among the agents induces a directed graph, where each node denotes an agent. An edge (u, v) exists if agent u values agent v's expertise, sociability, or both. This valued expertise or sociability may be in one or more domains. First, the graph structure is enhanced by adding labels to the edges of the graph, where the label on an edge (u, v) denotes agent v's sociability from agent u's point of view (in our notation, this is $\sigma_{u,v}$) for one domain. Agent u may model agent v's sociability for different domains. In other words, agent u might find agent v sociable for one domain, but not sociable for many other domains. The edges are labeled with the sociability values because sociability denotes how useful the other agent has been for locating services. Second, the sociability rank 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 agent i as a neighbor, N_j denotes neighbors of agent j, $\sigma_{j,k}$ denotes the sociability of k for j.

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

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

Example 2 Consider an agent j with neighbors i, k, and l such that $\sigma_{j,i} = 0.8$, $\sigma_{j,k} = 0.2$, and $\sigma_{j,l} = 0.2$. Then, j will contribute $\frac{0.8}{0.8+0.2+0.2} = 0.67$ to i's sociability rank, whereas only $\frac{0.2}{0.8+0.2+0.2} = 0.17$ to k and l each. ▮

The above 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.

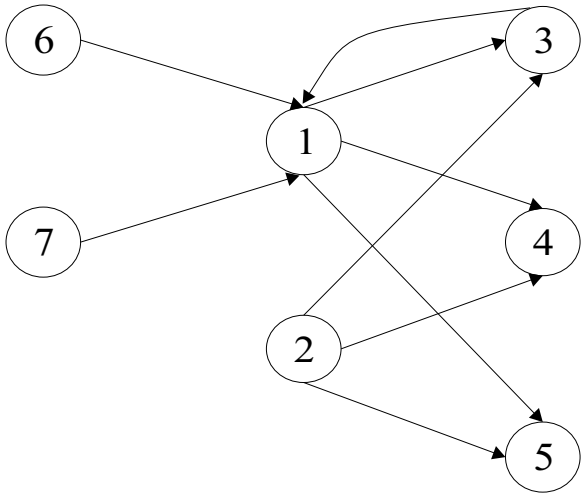

Figure 6: Agent 1 is ranked higher than agent 2

Example 3 The graph in Figure 6 shows part of a referral network, where agents 1 and 2 are pointing at the same set of agents 3, 4, and 5. However, agent 1 is being pointed to by more agents because it has been found sociable by more agents. Therefore, the ranking of agent 1 is higher than that of agent 2, which denotes that agent 1 is more in the community than agent 2. ▮

Whereas bipartite communities have a bipartite core, referral communities do not have any predefined structure. Unless the communities are mined, the members of the communities would not know the whole structure of the community. That is, each agent would follow referrals from its neighbors to locate useful agents. However, it would not specifically be able to identify the community from which it has benefitted. If information about each agent's neighbors is combined in a central repository, then the communities can be mined. Different mined communities may have different structures.

4.3 Correlation

Since the communities are targeted for locating services, the nodes with high hubness values are expected to be most useful to others. For this reason, we use the hubness values to rank the nodes of a bipartite community. While a bipartite community is a subset of the population, a referral community is a ranking of all the members of the population. When comparing a bipartite community with a referral community, the community size n is taken to be the size of the bipartite community found. The top n agents from the ranking of a referral community is then taken for comparison.

First, we calculate the correlation between communities found by both approaches using Spearman correlation, given in Equation 3 [Kendall, 1975]. Below, C and D denote two communities, consisting of the same members. 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.

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.

$$\rho(C, D) = 1 - \frac{6 \sum_{i=1}^{n} (C_i - D_i)^2}{n(n^2 - 1)} \tag{3}$$

To reduce manual work, we arbitrarily choose 10 communities for further comparison. The chosen communities vary in size such that the smallest community has 32 members and the largest community has 238 members (out of possible 400 agents). 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.

4.4 Utility

We evaluate the effectiveness of a community through its utility in finding service providers.

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]. Capability resembles cosine similarity but also takes into account the magnitude of the expertise vector. That is, expertise vectors with greater

magnitude are more capable for the given interest vector. In Equation 4, I ($\langle i_1 \dots i_n \rangle$) is an interest vector, E ($\langle e_1 \dots e_n \rangle$) is an expertise vector and n is the number of dimensions these vectors have.

$$I \otimes E = \frac{\sum_{t=1}^{n} (i_t e_t)}{\sqrt{n \sum_{t=1}^{n} {i_t}^2}} \tag{4}$$

Utility. The PageRank of a node depends only on its incoming edges, but ends up considering paths of length greater than one. In HITS, the hubness value of a node depends on its outgoing edges whereas the authority value of a node depends on its incoming edges. Although long undirected paths may be considered, the directed paths are limited to length of one. Hubs may have low authority values, and a hub that points at a hub gets credit only for the authority value of the hub being pointed at.

The *utility* of an agent denotes how easily it can access the information it needs. Like the HITS hubness value and unlike PageRank, the proposed utility metric considers the outgoing edges of an agent. Unlike HITS and like PageRank, it applies recursively. Unlike either of them, it is grounded via the capability metric introduced above. Further, whereas both HITS and PageRank ignore the strength of a link, utility works with edge weights, which capture the strength of a link. Edge weights are not obvious in traditional Web settings and traditional approaches (especially PageRank) implicitly assume that all outgoing edges are equally valuable. However, in a referral system, edge weights are learned by the agents and determine which edges would actually be followed.

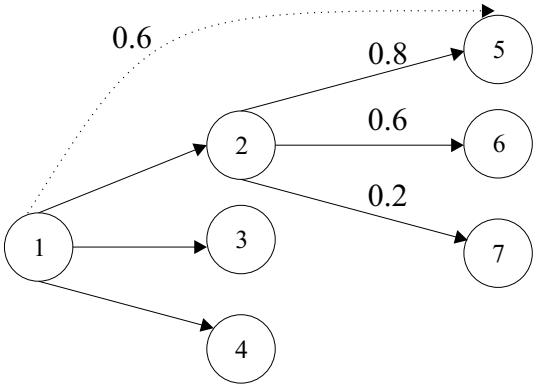

Figure 7: An example network labeled with utility values

Equation 5 is used to calculate the utility of an edge, $\mu_{i,j}$. 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, agent j will give a referral to the agent that provides most utility to agent j itself; i.e., the agent k that maximizes the utility of $\mu_{j,k}$. The usefulness of k for i is then calculated again by $\mu_{i,k}$. In other words, j will contribute to i by providing answers $(I_i \otimes E_j)$ or by giving a referral. The utility of an agent is then defined as the sum of the utilities of its outgoing edges.

$$\mu_{i,j} = \delta(I_i \otimes E_j) + (1 - \delta)\mu_{i,k} \tag{5}$$

where

$$k = \arg\max(\mu_{j,k}), \forall k \in N_j \tag{6}$$

Example 4 Consider the example in Figure 7 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. 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, $\mu_{1,2} = \delta(I_1 \otimes E_2) + (1 - \delta)\mu_{1,5}$. ∎

We compare referral and bipartite communities in terms of their total utility (Equation 5). Figure 8 gives a histogram of this comparison. The x axis shows the communities, labeled with letters (A through J). The y axis shows the total utility of the communities. 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 F receives a slightly worse utility than the bipartite community.

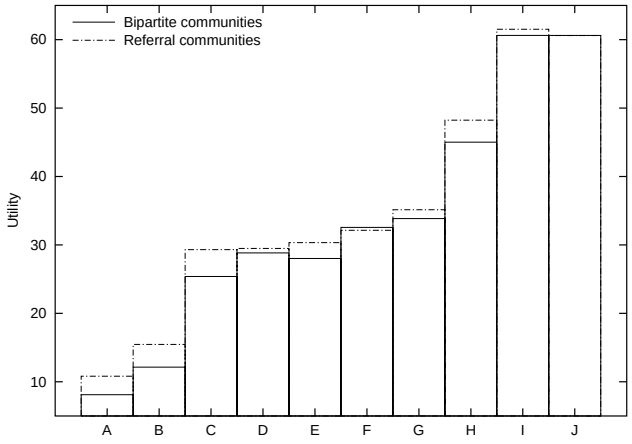

Figure 8: Comparison of utilities for some example communities

An agent with a high utility is either close to an expert, or close to another agent who will give a referral to the expert. Thus, the communities with higher utility can locate service providers more easily. Finding higher utility communities is especially important for exogenous applications [Domingos and Richardson, 2001]. For example, if a service provider is promoting a new product, the set of customers that are likely to use it are the ones that can actually locate the provider in the first place. Hence, the new product should be targeted to the community that yields the higher utility in terms of locating providers.

5 Discussion

The main difference between bipartite and referral communities lies in the importance assigned to referrals. Intuitively, it is only natural that referrals would prove valuable in a distributed environment. However, the practical payoffs are not obvious. This paper makes the following contributions. First, in Section 4.1, we find that when referrals are considered, more good answers are found. Second, in Section 4.4, based on the same motivation, we compare communities based on utility. Utility recursively captures the value of different members to each other. Because members in referral communities are selected based on their usefulness in providing answers or referrals, referral communities yield greater utility. Further, we develop an approach

to compare bipartite communities with referral communities. We also show that the communities are not correlated with respect to the importance they assign to different members.

Next, we review some related literature. In real life, people use referrals to seek information [Nardi et al., 2000]. This is an important motivation for referral systems. Multiple Intelligent Node Document Servers (MINDS) was the earliest agent-based referral system [Huhns et al., 1987]. Each node in the MINDS system is allocated a set of documents. Nodes help each other find documents in the network. Gradually, nodes learn how the documents are distributed in the network as well as the relevance preferences of individual users.

Kautz *et al.* develop ReferralWeb, an application that mines documents on the Web to uncover the social network of people [1997]. This social network is then used to find paths between any two people. Whereas the ReferralWeb models the relationships statically, we can model the relationship of agents dynamically by allowing them to adapt through neighbor changes.

Yu and Singh study referral networks in the context of scientific collaborations [2003]. They show that increasing neighbor set size and referral graph depth improve the accuracy of locating agents. Yu and Singh use weighted graphs to maintain referrals received for a single query. The nodes as well as the edges of the graphs are given weights, based on how valuable they are to the query originator. Yu and Singh develop an algorithm to minimize referral graphs such that agents follow only the most promising referrals.

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. 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 [2002]. Flake *et al.* define a web community as a collection of Web pages 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, some 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 in-degree) 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 constitute 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, although its connection to the community may be tenuous at best.

Alani *et al.* develop a method to identify communities of practice based on traversing ontologies [2003]. The relations in the ontologies are assigned a weight (automatically or manually) and the ontology is traversed for a specified number of links in the relations graph. Each node has a default weight that is updated based on the weights of its incoming edges. The nodes are then ranked based on their final weights.

The main advantages of our approach are that it seeks to capture intuitions behind communities in a direct manner, preserves the privacy of the participants, lets them to decide how to value endorsements, and enables endogenous applications. Our approach provides opportunities for further research. In future work, we plan to consider richer sociological ideas such as the ones in social networks as well as compare our approach to other definitions of Web communities.

6 Acknowledgments

This research was supported by the National Science Foundation under grant ITR-0081742. An earlier version of this paper appears in the IJCAI 2003 Workshop on Intelligent Techniques for Web Personalization. We thank Jiming Liu and the anonymous reviewers for helpful comments.

References

Harith Alani, Srinandan Dasmahapatra, Kieron O'Hara, and Nigel Shadbolt. Identifying communities of practice through ontology network analysis. *IEEE Intelligent Systems*, 18(2):18–25, March 2003.

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.

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.

Pedro Domingos and Matt Richardson. Mining the network value of customers. In *Proceedings of the 7th International Conference on Knowledge Discovery and Data Mining*, pages 57–66. ACM Press, 2001.

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.

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

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.

Henry Kautz, Bart Selman, and Mehul Shah. ReferralWeb: Combining social networks and collaborative filtering. *Communications of the ACM*, 40(3):63–65, March 1997.

Maurice G. Kendall. *Rank correlation methods*. Griffin, London, 4th edition, 1975.

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

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

Bonnie A. Nardi, Steve Whittaker, and Heinrich Schwarz. It's not what you know, it's who you know: Work in the information age. *First Monday*, 5(5), May 2000.

Sriram Raghavan and Hector Garcia-Molina. Crawling the hidden Web. In *Proceedings of the 27th International Conference on Very Large Databases (VLDB)*, pages 129–138. Morgan Kaufmann, 2001.

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

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

Munindar P. Singh. Deep Web structure. *IEEE Internet Computing*, 6(5):4–5, November 2002. Instance of the column *Being Interactive*.

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

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

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

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.

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.

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