University of Alberta

COMMUNITY MINING AND ITS APPLICATIONS IN EDUCATIONAL ENVIRONMENT

by

Reihaneh Rabbany khorasgani

A thesis submitted to the Faculty of Graduate Studies and Research
in partial fulfillment of the requirements for the degree of

Master of Science

Department of Computing Science

Fall 2010
Edmonton, Alberta

Examining Committee

Osmar R. Zaïane, Department of Computing Science

Denilson Barbosa, Department of Computing Science

Marek Reformat, Department of Electrical and Computer Engineering

Abstract

Information networks represent relations in data, relationships typically ignored in iid (independent and identically distributed) data. Such networks abound, like co-authorships in bibliometrics, cellphone call graphs in telecommunication, students interactions in Education, etc. A large body of work has been devoted to the analysis of these networks and the discovery of their underlying structure, specifically, finding the communities in them. Communities are groups of nodes in the network that are relatively cohesive within the set compared to the outside.

This thesis proposes Top Leaders, a fast and accurate community mining approach for both weighted and unweighted networks. Top Leaders regards a community as a set of followers congregating around a potential leader and works based on a novel measure of closeness inspired by the theory of diffusion of innovations.

Moreover, it proposes Meerkat-ED, a specific and practical toolbox for analyzing students' interactions in online courses. It applies social network analysis techniques including community mining to evaluate participation of students in asynchronous discussion forums.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

Data mining is recently challenged with finding patterns in structured and heterogeneous data. These structures are usually in the form of information networks, which encode the relations between data entities using graphs. This is in contrast with the traditional data mining approaches – *e.g.* association rule mining, supervised classification or clustering algorithms – which deal with independent and identically distributed data (IID) [21].

Neglecting the dependence structure of the data, and assuming the independence of data instances can lead to inappropriate conclusions [29]. For example considering only the content of web pages and overlooking their linking structure would lead to poor search results for a search engine. Therefore incorporating structural information of the data is crucial in pattern mining.

1.1 Thesis Motivations

Many application domains such as marketing, biology, epidemiology, sociology, criminology, and zoology produce inter-related data. These inter-relations could be represented using information networks while sharing a common trait, *i.e.* community structure. This structure refers to existence of groups as densely connected set of nodes when there is sparse connections between different groups. Detecting these communities is an important prerequisite to understanding of such structured data. To get a better sense of its practical importance, consider a group of web pages that have more links to each other than to the other pages. Being in the same group

may be linked to the closeness of topics, which might in turn enable search engines to better focus on a narrow set of related pages for answering queries [19].

This abundance of structured data has resulted in increasing popularity of community mining in social sciences such as psychology, anthropology, and criminology as well as in computer science and data mining. As of today, there has not been any consensus on the exact definition of community. For example, one approach defines it as a subgraph with density of inside edges greater than density of connections to the outside [24]. While another perspective, relies on the notion of structural similarity and further incorporates structural elements such as hubs and outliers [56]. Having these different definitions, many recent approaches have been proposed for finding communities in social networks [22, 12, 42, 11, 46, 56, 8, 9]. These approaches have promising results in some cases but still are not completely satisfactory and present some issues (described in detail in Chapter 2); such as assuming no prior or side information about the network; having problem in detecting highly inter-related or mixed communities; having problem in scalability for large networks.

In practice, we might have access to some prior information about the network in hand. As motivating examples, we see the number of communities in a blog network about US political elections, or in a business network, the analyst is only interested in top k company communities. Besides, these prior information could be obtained by exploiting the network using visualization systems [10]. This motivates development of methods utilizing available information about the network, such as the number of communities, to perform the community detection task more efficiently and accurately.

One the other hand, the interactions in online discussions forums is an interesting example of information networks. There is a growing number of courses delivered using e-learning environments and their online discussions play an important role in collaborative learning of students [17]. Even in courses with a few number of students, there could be thousand of messages generated in a few months within these forums. Unfortunately, current e-learning environments do not provide much information regarding the participation of students and the structure of interactions

between them. In many cases, only some statistical information is provided such as their frequency of posting [17]. Consequently, instructors have to monitor the discussion threads manually which is hard, time consuming, and prone to human error.

There is a recent line of work on applying social network analysis techniques to evaluate the participation of students in online courses [6, 15, 55, 34, 17]. While proposed methods tackled with some of its challenges, they do not address the problem completely (described in detail in Chapter 5). Therefore, it is interesting to investigate the practicability of social network analysis and community mining techniques in analyzing students interactions in online discussion threads.

1.2 Thesis Statments

This dissertation elaborates on the importance of social network analysis for mining structural data in the field of computer science and its applicability to the domain of education. More precisely it is addressing the following statements:

- **TS1:** Computing science could use theories of social network analysis from sociology to develop new methods for analysis of the stucture in relational data.
- **TS2:** Specifically, diffusion of innovation theory which describes the structural features that influence whether individuals will join communities, could be useful in measuring the closeness of nodes in structured data.
- **TS3:** The widely used k-means algorithm in clustering could be an inspiration for an analogous and effective approach in community mining.
- **TS4:** Social network analysis could have useful applications in e-learning — for monitoring and evaluating participation of students in online courses.

1.3 Thesis Contributions

In this dissertation, we present a new community detection approach based on finding top-k leaders in an information network. In contrast to other methods, we exploit prior knowledge about a given network, such as the desired number of commu-

nities to be found. In our work, we assume that each community has a representative leader node, which is the most central node in that community. A community is a set of all follower nodes assembling close to a leader. Briefly, our approach first finds promising leader nodes in the given network, then iteratively updates communities and their corresponding leaders until there is no change in the communities. This scheme is very similar to the partitioning philosophy adopted in clustering methods such as k-means. While k-means is infamously sensitive to noise, our approach, Top-Leaders, coupled with our notion of closeness highlighted herein, allows us to identify marginal nodes in a network as outliers and thus is not affected by noise. Moreover, hubs, nodes that connect different communities, can also be identified. The closeness measure we propose, Intersection Closeness (*iCloseness* for short), is a novel measure to assess the relations between community nodes. We use this measure to both find the initial leaders in a network as well as to update relations among community nodes. For instance, in the initialization we choose potential leaders who are not too *iClose* to each other to avoid starting with leaders that might end up within the same true community. When associating followers with leaders, we assign a follower to the *iClosest* leader.

This closeness measure encapsulates the notion of membership in a community and its basic idea is reinforced by observations made on community dynamics in social networks with regard to the probability of joining a group based on the concept of diffusion of innovation [3]. It is observed that the likelihood of joining a community in social networks depends upon the number of pre-existing connections with group members and the density of edges between these members and other members in the group. In other words, if I am faced with two groups in which I already have friends and if I need to join one of these groups, I could choose either one, but there is a higher probability to join the group in which I have more friends. In addition if I am faced with two groups in which I have the same number of friends, there would be a higher probability that I would join the group in which the connectivity of my friends with the group is stronger.

Moreover, we proposed Meerkat-ED, a specific and practical toolbox for analyzing students interactions in asynchronous discussion forums of online courses.

It analyzes the structure of these interactions using social network analysis techniques including community mining. Meerkat-ED prepares and visualizes overall snapshots of participants in the discussion forums, their interactions, and the leaders/peripheral students in these discussions. Moreover, It creates a hierarchical summarization of the topics discussed in the forums, which gives the instructor a quick view of what is under discussion in these forums. It further illustrate how much each student has participated in these topics, by showing his/her centrality in the discussions on that topic, the number of posts, replies, and the portion of terms used by that student in the discussions.

1.4 Thesis Organization

This dissertation is organized into two parts. The first part introduces community mining in social networks and our new approach for detecting communities. The second part illustrates the practicability and practicality of social network techniques including community mining in analyzing interactions of students in online discussions and presents our specific social network toolbox for such analysis. Following these two parts, Chapter 7 concludes the overall contributions of this thesis.

The first part is divided into three chapters; Chapter 2 briefly introduces social networks and social network analysis and then surveys the current approaches for community mining in social networks. The next chapter, Chapter 3, details our iCloseness measure and proposes the community detection algorithm, named Top Leaders. And finally Chapter 4 reports the result of our experiments on accuracy and efficiency of Top Leaders approach for both real and synthetic benchmarks compared to other state-of-the-art contenders.

The second part of this dissertation has two chapters. The first chapter, Chapter 6, illustrates the place and need for social network analysis in study of the interaction of users in e-learning environments and then summarizes some recent studies in this area. The following chapter, Chapter 6, presents Meerkat-ED – our solution for social network analysis of online courses – and illustrates its practical application on our own case study data.

Part I

Top Leaders: Community Detection based on Diffusion of Innovation

Chapter 2

Background and Related Works

Social life is relational; it's only because, say, blacks
and whites occupy particular kinds of patterns
in the networks in relation to each other that
"race" become an important variable.
Collin (1988)

2.1 Social Networks

First introduced in social and behavioral sciences and focused on relations between entities and patterns of these relations, social networks are formally defined as a set of actors or network members whom are tied by one or more type of relations [38].

The actors are most commonly persons or organizations however they could be any entities such as web pages, countries, proteins, documents, etc. and sometimes under a more general name, information networks. There could also be many different types of relationships, to name a few, collaborations, friendships, web links, citations, information flow, etc. [38]. These relations represented by the edges in the network connecting the actors and may have direction (shows the flow from one actor to the other) and strength (shows how much, how often, how important).

2.2 Social Network Analysis

Unlike individualist or scientists in attribute based social sciences, social network analysts argue that causation is not located in the individuals, but in the social structure [38]. Social network analysis is the study of this structure.

Rooted in sociology, nowadays, social network analysis has became an interdisciplinary area of study, including researchers from anthropology, communications, computer science, education, economics, criminology, management science, medicine, political science, and other disciplines [38]. For example in medicine, it is used to understanding the progression of the spread of an infectious disease [31], in criminology, it is an important part of a conspiracy investigation and identifying the nature and extent of conspiratorial involvement [14], or in education it is helpful in monitoring interactions and participation of students in online courses [44].

Social network analysis examines the structure and composition of ties in the network to answer questions like:

- **Prestige:** Who are the central actors in the network?
- **Influence:** Who has the most outgoing connections?
- **Prominence:** Who has the most incoming connections?
- **Outlier:** Who has the least connections?
- **Density:** What proportion of possible ties does actually exist?
- **Path Length:** How many actors are involved in passing information through the network?
- **Community:** Which actors are communicating more often with each others?
- **etc.** ...

The question we are focused in this part of the thesis is how one can find the communities in a given social network. Communities are cohesive subgroups of actors among whom there are relatively strong, direct, intense or frequent ties [54]. In the rest of this chapter, we first list the important terms in social network analysis and then summarize the related work to community detection in social networks. We propose our new community detection approach in the subsequent chapters.

- **Neighbourhood** of a node consists of its adjacent nodes i.e. the nodes directly connected to it.
- **Bridge** is defined in graph theory and is referred to as an edge connecting different components of a network i.e. removing it from the network would increase the number of connected components in the network.

- **Clique** is a subgraph of a network in which all the actors are connected to each other.
- **Centrality** is a notion of prominence or social power in networks [38] and has different indices:
  - **Degree centrality** shows the number of ties a node has to the other actors of the network. In case of directed graphs, it could be divided into indegree and outdegree centrality [38]. More formally, the degree centrality of node n with degree $deg(n)$ – its number of adjacent edges – is computed as: $C_D(n) = \frac{deg(n)}{N-1}$, where N is the size of the network.
  - **Betweenness centrality** shows the influence of a node (or an edge) on the flow of information between other members in the network. It is computed based on the number of shortest paths runs through that node (or edge) [42]. Let σ_{uv} denotes the number of shortest paths between node u and v and $\sigma_{uv}(n)$ denotes the number shortest paths between u and v that runs through n. The betweenness centrality is computed as: $C_B(n) = \sum_{u,v \neq n} \frac{\sigma_{uv}(n)}{\sigma_{uv}}$.
  - **Closeness centrality** shows how close a node is to other members of the network. It is computed based on the length of shortest paths from this node to all the other actors in the network [38]. Assuming $d_\sigma(n, v)$ denotes the length of the shortest path from node n to node v, the closeness centrality of node n is defined as: $C_c(n) = \frac{1}{\sum_v d_\sigma(n,v)}$.
  - **Eigenvector centrality** shows how close a node is to other powerful members in the network. It is computed by assigning relative scores to nodes based on scores of their neighbours. PageRank is a well-known example of this measure.
- **Density** is a measure for how connected the actors are in a network i.e. the proportion of ties in the network divided by the total number of possible ties.
- **Cohesion** is the extent to which the actors are tied with nodes in their subgroup rather than rest of the network [38].

2.3 Community Mining in Social Networks

Community detection in social networks has been pursued by sociologists for many decades. More recently, it has also attracted attention from physicists, applied mathematicians and computer scientists [40]. The availability and growth of large datasets of information networks makes community mining a very popular research topic in computing science. This line of research resembles well-studied clustering methods in machine learning. However, clustering approach in machine learning is closer to individualist approach in social sciences, as they both use the *attributes* of data entities. This is in contrast with social network analysts' perspective which is more focused on the *relation* between the entities. This view seems closer to graph partitioning problems in machine learning. Unfortunately one may not apply available methods for parititioning to the problem of community mining because of the assumptions on availability of predefined partition size. which is not a valid assumption for real social networks. [41].

2.3.1 Graph partitioning Approaches

Graph partitioning is a traditional and well-studied approach in parallel computing, circuit partitioning and layout. The objective of graph partitioning algorithms is to divide the vertices of the graph into k groups of predefined size, while minimizing the number of edges lying between these groups, called *cut size* [20]. Finding the exact solution in graph partitioning is known to be NP-complete. However there are several fast but sub-optimal heuristics such as METIS [30], flow-based methods [19], information-theoretic methods [16] and the most well-known Kernighan-Lin algorithm [32].The Kernighan-Lin algorithm, designed primarily for circuit layout, optimizes a benefit function Q, which represents the difference between the number of edges inside the groups and the number of edges running between them. It start by partitioning the graph into two groups of equal size and then swaps subset of vertices between these groups to maximize Q. After a series of swaps, it chooses the group with largest Q for partitioning in the next iteration. This algorithm has $O(n^2 \log n)$ time complexity where n is the number of vertices [20].

Another important family of graph partitioning algorithms is the spectral clustering methods [43]. They divide the network into groups using the eigenvectors of matrices, mostly the Laplacian matrix. For example the spectral bisection method, divides the graph by repeated bisection. In every step, the cut size of a partition in two groups is $R = \frac{1}{4}s^T Ls$, where L is the Laplacian matrix and s is a vector showing the partition: s_i is either $+1$ or -1 that shows to which group the corresponding vertex belongs. It could be shown that $R = \sum_i a_i^2 \lambda_i$ where λ_i is the i^{th} eigenvalue of L. Therefore for minimizing the R, it finds the first non zero eigenvalue of the Laplacian matrix and having its corresponding eigenvector, the vertices would be partitioned into two groups, where vertices with positive element in the eigenvector are placed in one group and negative ones in the other [20].

The major incompatibility of these methods is that *community structure detection* assumes that the networks divide *naturally* into some partitions and there is no reason that these partitions should be of the same size. In minimum cut methods, it is necessary to specify both the number of groups and the size of the groups. If one tries to minimize the cut size without fixing the number of groups, the solution would be grouping all the vertices in one group. Likewise for the size of the groups, just minimizing the cut size would result in separating the vertex with lowest degree from the rest of the graph as a group.

Several alternatives measures have been proposed for the cut size, such as *conductance* [5] *ratio cut* [7] and *normalized cut* [50]. The conductance of the subgraph $\mathcal{C}$ is defined as

$$\Phi(\mathcal{C}) = \frac{c(\mathcal{C}, \mathcal{G}\backslash\mathcal{C})}{min(k_{\mathcal{C}}, k_{\mathcal{G}\backslash\mathcal{C}})}$$

Where $c(\mathcal{C}, \mathcal{G}\backslash\mathcal{C})$ is the cut size of $\mathcal{C}$ and $k_{\mathcal{C}}$, $k_{\mathcal{G}\backslash\mathcal{C}}$ are respectively the sum of degrees of vertices in $\mathcal{C}$ and outside $\mathcal{C}$. This would result in picking groups with nearly equal total degrees and hence approximately equal size [20]. The ratio cut is defined as

$$\Phi(\mathcal{C}) = \frac{c(\mathcal{C}, \mathcal{G}\backslash\mathcal{C})}{min(n_{\mathcal{C}}, n_{\mathcal{G}\backslash\mathcal{C}})}$$

where $n_{\mathcal{C}}$, $n_{\mathcal{G}\backslash\mathcal{C}}$ denotes the number of vertices in the subgraphs. The normalized cut is also defined as

$$\Phi(\mathcal{C}) = \frac{c(\mathcal{C}, \mathcal{G}\backslash\mathcal{C})}{k_{\mathcal{C}}}$$

These two measures also biased towards divisions into equal-sized groups [20].

The assumption or bias of equal-sized groups, along with the fact that most of these graph partitioning methods are iterative bisectioning, makes them inappropriate for community detection. The later is a problem because for example a partition of three groups would be obtained by splitting one of the partitions from the first bi-partition while the optimal solution could be a group containing vertices from both of those groups. The next class of approaches, Hierarchical Clustering approaches, are more desirable in community detection which are described in the next section.

2.3.2 Hierarchical Clustering Approaches

Hierarchical clustering approaches define a similarity measure between vertices and group similar vertices together to discover the natural divisions in a given network. Hierarchical clustering approaches classify into two major categories: agglomerative algorithms and divisive algorithms. Agglomerative algorithms iteratively merge groups with high similarity, while divisive algorithms iteratively remove edges connecting vertices with low similarity. In agglomerative methods, the similarity between vertices needs to be generalized for groups. Which could be done in three ways: single linkage, complete linkage or average linkage. In single linkage, the similarity of two groups is defined as the most similar vertices between those groups, while the complete linkage defines it as the least similar vertices. And average linkage computes the average similarities of every pairs of vertices in two groups as the similarity of those groups.

The result of hierarchical approaches depend on the similarity measure they use. There are many similarity metrics proposed to measure similarity between vertices, mostly by measuring the similarity of neighbourhood of those vertices or their structural equivalence using their corresponding rows of the adjacency matrix (A). For example the *Euclidean distance* [54] computes the similarity of vertex i and j as follows:

$$d_{ij} = \sqrt{\sum_{k \neq i,j} (A_{ik} - A_{jk})^2}$$

Alternatively one could measure the similarity using the *Pearson correlation* [20]

between rows of the adjacency matrix which is defined as:

$$C_{ij} = \frac{\sum_k (A_{ik} - \mu_i)(A_{jk} - \mu_j)}{n\sigma_i\sigma_j},$$

$$where \quad \mu_i = \frac{\sum_j A_{ij}}{n} \quad and \quad \sigma_i = \sqrt{\frac{\sum_j (A_{ij} - \mu_i)^2}{n}}$$

Another common measure computes the similarity by measuring the overlap between neighbourhoods $\aleph(i)$ and $\aleph(j)$ of the vertices i and j [20] using the *Jaccard index*:

$$w_{ij} = \frac{|\aleph(i) \cap \aleph(j)|}{|\aleph(i) \cup \aleph(j)|}$$

These hierarchical approaches do not presume the size of groups, however they have other drawbacks in the context of community mining. The major weakness of agglomerative methods is their time complexity ($O(n^2)$ for single linkage and $O(n^2 \log n)$, overlooking the calculation time of the chosen similarity measure) which makes them unscalable for large networks. Moreover, they tend to miss the periphery vertices due to their low similarity. On the other hand the divisive methods, tend to cluster all the periphery nodes, even the outliers, and often as a separate clusters. Besides, hierarchical approaches in most cases generate an artificial hierarchy while do not provide a way to choose which one of these partitioning levels represents the real community structure of the network [20].

GirvanNewman algorithm

Girvan and Newman [22] proposed the first community detection approach using the social network analysis techniques and opened a new venue for community detection algorithms. Their method is a divisive hierarchical clustering algorithm which iteratively removes the edge with highest betweenness to obtain the community structure of the network. The betweenness of an edge could be computed as the number of shortest paths running through that edge. High betweenness is a sign for bridges in the network, which are edges connecting different communities, illustrated in Figure 2.1. Their approach obtains good result in real world data sets, however, it is computationally expensive and unscalable for large networks as it

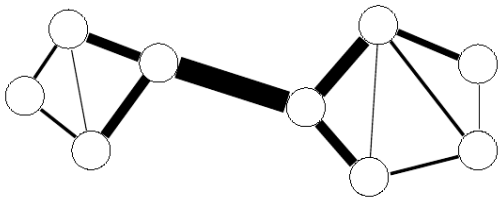


Figure 2.1: Edge Betweenness: edges connecting different communities have high betweenness. In this figure, the thickness of edges represents their betweenness. The edge between two communities, has the highest betweenness as all the shortest paths between any pair of vertices, which are in different communities, have to run through this edge.

needs running time of $O(m^2n)$ in general and $O(n^3)$ in sparse networks, where m is the number of edges [22].

For choosing which level of hierarchy best represents the community structure of the network, in [42], Newman proposed modularity Q, which is a measure for the quality of a particular division of a network. This measure became an objective for a class of popular approaches that try to maximize the modularity for finding good communities. The following section further elaborates on these approaches.

2.3.3 Modularity Based Approaches

The modularity (Q) is a measure for assessing communities which shows the quality of that particular partitioning of the network. Its basic idea is to compare the partitioning with a randomized network with exactly the same vertices and same degrees, in which edges are placed randomly regardless of community structure. It measures how well the edges fall within the communities compared to the randomized network. Let A be the adjacency matrix of the network containing m edges ($m = \frac{1}{2}\sum_{ij} A_{ij}$); then the portion of edges within communities is:

$$\frac{1}{2m}\sum_{ij} A_{ij}\delta(R^i, R^j)$$

Where R^i shows the community that vertex i belongs to, and $\delta(x, y)$ is one if x is equal to y and zero otherwise. The modularity is then defined as subtraction of this quantity from its expected value for the randomized network (with same nodes

where edges created randomly but with respect the node degrees).

$$Q = \frac{1}{2m} \sum_{ij} [A_{ij} - \frac{d_i d_j}{2m}] \delta(R^i, R^j)$$
$$where \quad d_i = \sum_j A_{ij}$$

There are variety of approximate optimization algorithms for searching the space of possible partitionings of a network in order to detect the partitioning with the highest modularity. Newman uses a greedy optimization in [42]. He employs an agglomerative clustering method, starting with each vertex as a cluster, in each step it merges the clusters that most increase the modularity. Later Clauset [12] presented a very fast version of the algorithm , called *FastModularity*, which has become very popular ever since.

There are some doubts in the usefulness of modularity. In [23], Good et al. have shown that on some real networks, the communities corresponding to the optimal modularity fundamentally disagree with the ideal communities. We also observed the same phenomenon in our results as described in Chapter 4.

2.3.4 Other Approaches

In addition to the prominent Q-modularity approach [12] mentioned earlier – against which we compare our results – there are two other algorithms worth mentioning that are not only innovative in the process of discovering communities but are also highly effective in many cases: CFinder [46] and SCAN [18]. We also compare our results against these two contenders.

Clique Percolation

Clique Percolation method, called CFinder, is proposed by Palla et al. [46] to partition networks into overlapping communities. Based on the observation that edges within communities are likely to form cliques, they defined a community as union of adjacent cliques. More precisely, they used the term k-clique for a complete subgraph of size k, considering two k-cliques adjacent if they share $k-1$ vertices. They defined a k-clique community as the largest connected subgraph obtained by union of k-cliques which are reachable from each other through a series of adjacent

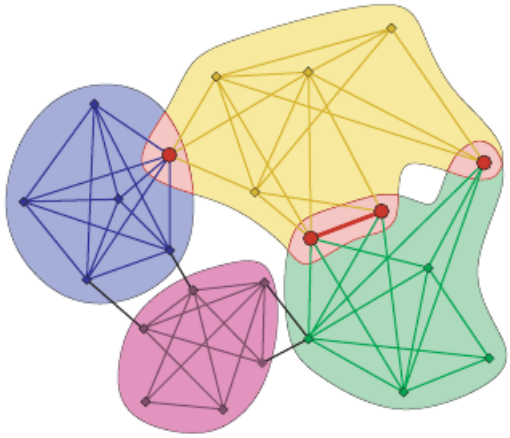


Figure 2.2: Clique Percolation: an example illustrating the detected communities by clique percolation with k equal to 4. The red vertices are the overlapping ones which belong to more than a community. K-cliques (complete subgraphs of size k) are reachable only by other k-cliques from their community, where k-cliques are adjacent if they share $k-1$ vertices. Figure is reprinted from [46]

k-cliques. Here k is an input parameter for the method where k between 3 and 5 obtains very good results on real world networks [46]. Since a vertex could belong to more than one clique, it could also belong to different communities detected by this method, which enables detection of overlapping communities. Figure 2.2 illustrates an example of detected communities using this method.

SCAN

Rooted in the well known density-based clustering algorithm DBScan [18], Xu et al. derived a similar approach for community detection in social networks and proposed the SCAN algorithm [56] which detects not only communities, but also hubs and outliers in networks. Hubs are actors who have ties with many communities but belong to none, and outliers are actors who are outsiders and do not belong to any community. Similar to the notion of reachability in DBScan, SCAN uses the neighbourhood of a vertex for community mining. Nodes that are structurally reachable from each other are grouped together in the same community. More specifically, *structural similarity* for vertices i, j with immediate neighborhood $\aleph(i)$, $\aleph(j)$ is

defined as:

$$\sigma_{ij} = \frac{|\aleph(i) \cap \aleph(j)|}{\sqrt{|\aleph(i)||\aleph(j)|}}$$

Having the similarity of every connected pair of vertices, they defined the ε-neighbourhood of a vertex as the vertices with similarity higher than ε with that vertex. They further considered the vertices that have ε-neighbourhood of a size greater than μ as the cores, and defined a community as vertices that are reachable to each other through a set of cores, while reachability to a core is being in its ε-neighbourhood.

Their performance appears to be very good but it is highly dependant on their two parameters: the structural similarity threshold for a "core" vertex, ϵ, and the minimum number of neighbours needed to propagate the reachability, μ. This sensitivity to parameters could be addressed using a visual data mining approach [10].

Recent Directions

The more recent directions in community mining includes the investigation of local algorithms for very large networks, the detection of overlapping communities, and the examination of community dynamics. When networks are too large to realistically fit in main memory such as the entire World Wide Web, approaches that consider global information about the network are inadequate and local methods are unavoidable [9, 11, 36]. These methods use local information by expanding the neighbourhood around a given node or set of nodes to identify communities that encompass the starting nodes in question. Fuzzy methods [24, 39, 46, 58] allow nodes to belong to multiple communities. Indeed, many real world networks present genuine overlap between true communities.

Similar to the interest in studying cluster changes in time or clustering data streams, there is new research interest in investigation of the dynamics of group formation in social networks [4, 53, 2]. Backstrom et al. [3] studied the evolution of large-scale social networks over time and found that the tendency of an individual to join a community is influenced by his number of friends in that community and also crucially by how those friends are connected to one another. This is the source of inspiration for our new closeness measure presented in the following chapter.

Chapter 3

Top Leaders Approach

3.1 Motivations

In this chapter we present in detail a new approach for detecting communities in social networks, named Top Leaders. Top Leaders is inspired by the well-known k-medoids clustering algorithm; however it works based on the relations between data points instead of their attributes. Similar to principal, this algorithm consists of choosing k representative nodes as leaders and then associating other nodes, the followers, to one of these leaders based on the relations/links between nodes to form communities. It iteratively elects new leaders for each community and reassigns nodes to the leaders to form new communities. Convergence is attained when the best leaders are found and each node is associated to its most appropriate leader.

Similar to k-medoids, our algorithm is sensitive to its initialization, the selection of the initial k leaders. We have experimented with a variety of strategies, from a random selection à la k-means to more advanced heuristics. And we came up with an adjustable initialization method which could be adjusted with the prior knowledge of the network and its communities to yield very accurate results.

Top Leaders works based on a new measure of closeness, *iCloseness*, inspired by the theory of Diffusion of Innovations. This closeness measure is computed based on the intersection of neighbourhoods and quantifies the closeness between a node and a leader (i.e. most central node of the community). We use *iCloseness* in associating nodes to communities and in selecting initial leaders.

In the following, we first introduce *iCloseness*, then we describe the general

framework of our algorithm and our initialization methods. After that we elucidate the processes associating followers to a leader, electing new leaders, and detecting outliers. Finally, we present a generalization of the algorithm for weighted networks.

3.2 Measuring Closeness

Top Leaders assumes that a community is constituted of a leader and the follower nodes associated to it; where the community leader is the most central member in its community. With leaders representing communities, the community membership of the remaining nodes is the association of followers to nearby leaders. This association is very much related to the theory of *Diffusion of Innovations* and its application to information networks [51, 49].

Diffusion of Innovations which stems from research in sociology, is a theory of how, why, and at what rate new ideas and technology spread through cultures. Specifically for the case of information networks, if we consider the act of joining a community as a behaviour that spreads within a network, then based on this theory the probability of joining a community depends on the number of friends one already has in the community and the internal connectedness of the friends within. This is the main idea behind our closeness measure, the Intersection Closeness (*iCloseness*), which is based on the common neighbours between two nodes within a predefined neighbourhood. This concept of the increase of the probability of joining with the increase of existing friends in a community and their connectivity is argued in [3] in the context of social networks dynamics. Socially, there is indeed advantage in joining a group with friends that know each other and who are connected.

To measure *iCloseness* of leaders and a given node n, we compute the intersection of these leaders' neighbourhoods and n's neighbourhood, i.e., how many neighbours they have in common (Figure 3.1a). One would join the community in which there are more friends already in. The density of the intersection is also considered. One is more tempted to join a group where he or she has friends who

already know each other (Figure 3.1b) and are connected within the community (Figure 3.1c).

More formally, let $\aleph(n, d)$ denote the neighbourhood of depth d for node n (n itself is included), which is an induced subgraph of the network, formed by all nodes reachable from n by traversing at most d edges. Assuming $|S|$ shows cardinality (size) of set S, we define the ***iCloseness*** as follows:

$$iCloseness(n_1, n_2, d) = |S| + Density(S) \tag{3.1}$$
$$where \quad S = \aleph(n_1, d) \cap \aleph(n_2, d)$$

The density of an intersection is the proportion of edges in the intersection relative to the total number of possible edges within it. Let A be the adjacency matrix of the network where $A_{ij} = 1$ if vertex i is connected to vertex j and 0 otherwise, then the density of the subgraph S representing the intersection is obtained by:

$$Density(S) = \frac{\sum_{i,j \in S} A_{ij}}{|S| \times (|S| - 1)} \tag{3.2}$$

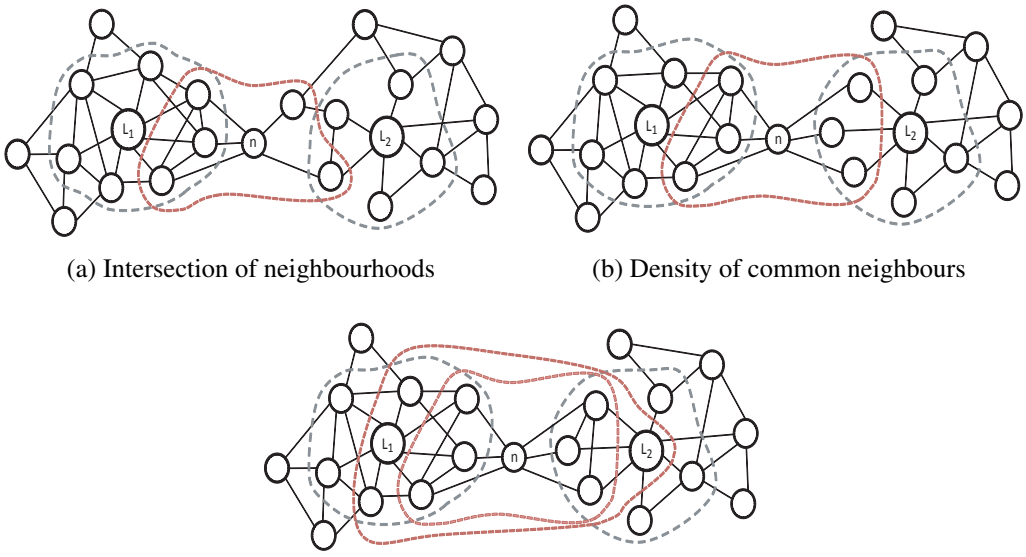
Note that ***iCloseness*** is symmetric but does not satisfy the triangle inequality.

3.3 Main Framework

The basic idea of the Top Leaders algorithm, is first to find some k community leaders, and then determine the community membership of other nodes in the network based on their relations to the identified leaders. This relationship is based on a notion of closeness, clarified in Section 3.2, which in turn stems from observations made on the dynamics of group formation in social networks.

Algorithm 1 highlights the major steps of Top Leaders algorithm. The first step is the selection of the initial k leaders which is described below in Section3.3.1. The second step is an iteration in which we alternate between association of followers and election of new leaders. First, nodes are either associated to a leader or labeled as outliers (elaborated further in Algorithm 2), and second, when all nodes in the network are dealt with, a new leader is picked in each community.

Below, we focus on the initialization of leaders and on how to associate nodes



(a) Intersection of neighbourhoods

(b) Density of common neighbours

(c) Expanding Neighbourhoods

Figure 3.1: Determining community of node n: n should be assigned to leader L_1 because **a)** n has more common neighbours with L_1 than L_2, **b)** although n has the same number of common neighbours with L_1 and L_2, its common neighbours in L_1's intersection are more connected to each other, **c)** although n has the same number of common neighbours with L_1 and L_2 and both intersections are equally dense, it has more common neighbours with L_1 if we expand its neighbourhood boundary by one.

Algorithm 1 Top Leaders algorithm

Input: Social network G, integer k

```
initialize k leaders
repeat
  {finding communities}
  for all Node n ∈ G do
    if n ∉ leaders then
      associate n to the iClosest leader {Algorithm 2}
    end if
  end for
  {updating leaders}
  for all l ∈ leaders do
    l ← arg max_{n ∈ Community(l)} Centrality(n)
  end for
until there is no change in the leaders
```

to leaders. Since the initialization also exploits the notion of closeness used in associating followers to leaders, we start by clarifying our *iCloseness* measure.

3.3.1 Initialization Methods

Like with any partitioning process, the initialization is crucial. Starting with the correct leaders allows quick convergence while starting with the wrong leaders will necessitate many iterations and may get stuck in a bad local optimum. This is a known problem with k-means for instance and many ways out were suggested in the literature such as running the nondeterministic algorithm multiple times or suggesting heuristics for the selection of better starting points. We experimented with myriad strategies specific to information networks and report here some initializations from a naïve random selection of leaders to a more elaborate approach. We highlight herein these initialization methods and show their impact in the experiments in Chapter 4.

Naïve Initialization

The naïve initialization is a random selection of k nodes from the network. This is simple but is not deterministic and may lead to bad results.

Top Global Leaders

Founded on the fact that community leaders are central nodes in their community, this initialization method picks the k most central nodes in the network as the initial leaders. This approach is also naïve because while community leaders are indeed central in their respective communities there is no reason that they should be the most central in the global network. In fact this methods produces good results in some cases but it yielded results worse than random in some others. On average, however, this strategy seems satisfactory. We added a variation such as selecting randomly k leaders from a larger set ck of most central nodes in the graph where c is a constant. However, this did not produce better results and in addition is non deterministic.

Top Leaders & not Direct Neighbour

The major drawback of Top-Global-Leaders is the fact that it is possible that two of the most central nodes belong to the same community. Choosing arbitrarily the k most central nodes may force a community to split and this would negatively affect the final results. Therefore we propose to choose the k most central nodes that are not directly connected to each other which avoids choosing leaders in the same community. To implement this strategy we start from the most central node, and add the next central one to the current set of leaders if it is not directly connected to any of the already selected leaders.

Top leaders & not iClose

Top Leaders & not Direct Neighbour method improves the result significantly but still produces inaccurate results in some cases. This intermittent inaccuracy is due to the fact that being direct neighbours does not exclude being in different communities. Therefore, the method could occasionally mistakenly avoid choosing two correct leaders that are directly connected but truly in different communities. To steer clear from this problem, instead of using a simple direct connectivity, we use our already defined *iCloseness* to measure how close a node is to a given leader (i.e. how much it belongs to that leader's community). The computation is simply after starting with the most central node, we add the next central one to the current set of leaders after checking its *iCloseness* to the already selected leaders and add it to the current set of leaders if it is not too *iClose* to any of the already selected leaders, compared to a threshold. In our experiments, this threshold is set 5 while this value was tested and is stable with most networks we encountered.

3.3.2 Association of Nodes to Leaders

Algorithm 2 depicts the process of associating a node to its *iClosest* leader. For finding the *iClosest* leader for a given node, we initialize its candidate leaders by considering all the leaders in its view, $\aleph(n, 2 \times \delta)$, these might possibly have common neighbours of depth δ with that node. We start measuring *iCloseness* between the node and its candidate leaders, by neighbourhood depth 1 (which consists of the

nodes that are directly connected to this node). If there is more than one candidate leader (with the maximum *iCloseness*) for this node we would expand the node's neighbourhood by one (by adding nodes that are directly connected to the current nodes in its neighbourhood). We keep expanding the neighbourhoods as long as there are ties up to the neighbourhood depth threshold (δ).

Algorithm 2 Associate n to the *iClosest* leader

Input: Social network G, node n, set of k leaders

depth $\leftarrow$ 1
CanList $\leftarrow$ leaders $\cap$ $\aleph$(n, 2$\times\delta$)
repeat
  CanList $\leftarrow \underset{\substack{\text{c} \in \text{CandList} \wedge \\ \text{iCloseness(n,c,depth)} > \gamma}}{\arg\max}$ iCloseness(n, c, depth)
  depth $\leftarrow$ depth+1
until |CanList| $\leq 1 \vee$ depth $> \delta$
if |CanList| $= 0$ **then** {No candidate leader}
  associate n as an outlier
else if |CanList| > 1 **then** {Many candidates}
  associate n as a hub
else {Only one candidate leader in CanList}
  associate n to CanList
end if

The algorithm is not sensitive to the value of δ. We have experimented with a variety of networks and different depth grater than 2 have given the same result. Thus, we set the δ threshold at 2.

3.3.3 Updating Leaders

The reassignment of leaders is simply the election of the node with the highest centrality in a community ($\arg\max_{n \in Community(l)} Centrality(n)$). This is because the centrality of nodes in a community measures the relative importance of a node within that group. For computing the centrality we use a generalization of degree centrality which also works for weighted networks. The degree centrality for a node n within a community is the number of edges from the community incident upon n and represents to some extent the "popularity" of n in the community. For a community C of size N, the degree centrality of a node n in C is $DC(n) = \frac{deg(n,C)}{N-1}$

where $deg(n, C) = \sum_{m \in C} A_{nm}$, is the number of edges in C incident upon n.

3.4 Outlier Detection

To detect outliers in the network, we define an outlier threshold (γ). Only leaders that are *iCloser* than this threshold to the node are considered. If after reaching the neighborhood threshold, the node is still not *iClose* enough to any of the current leaders; it is marked as an outlier. Hubs are those nodes that follow more than one leader. They sit on the intersection of communities. Different value of the γ outlier threshold can give different results. When we know that no outliers exist in the network, γ is set to 0. Otherwise γ depends on the density of the network, and to correctly identify outliers, γ could vary between 1 and 4 in most cases.

3.5 Weighted Top Leaders

For weighted networks, we should generalize the notion of *iCloseness*. We define the *Belongness* function B that represents to which degree a node belongs to a leader. Belongness of nodes in the neighbourhood of leader l is calculated while expanding the neighbourhood of l as described in Algorithm 3. Note that if the network is unweighted this value would be always one for all neighbours of a leader. Having this function, we redefine the *iCloseness* as follows:

Algorithm 3 Calculating Belongness

Input: Social network G, node l

```
B[l,l] ← 1
for depth = 0 to δ do
  for all n ∈ ℵ(l, depth) do
    for all m incident to n do
      B[m,l] ← max(B[m,l], B[n,l] × W_mn)
    end for
  end for
end for
```

$$iCloseness(n_1, n_2, d) = \sum_{v \in S} B(v, n_1) \times B(v, n_2) + Density(\aleph(n_1, d) \cap \aleph(n_2, d)) \tag{3.3}$$

Similarly the Density is also generalized for the weighted graph. Let W show the weight matrix of the network (after normalization where all weights are between 0 and 1) which would be equal to the adjacency matrix if the network is unweighted, then the density of the weighted subgraph S is obtained by:

$$Density(S) = \frac{\sum_{i,j \in S} W_{ij}}{|S| \times (|S| - 1)} \tag{3.4}$$

Apart from *iCloseness* we need to have a minor change in the process of updating leaders. For computing the degree centrality for a node in a weighted network, we sum up the (normalized) weights of edges from the community incident upon n instead of simply counting them. Consequently, for comparing the nodes in each community and selecting the node with the highest centrality as the new leader, we compare them by $\sum_{m \in C} W_{nm}$ (note that we do not need the division as the denominator is the same for all nodes in the community).

Chapter 4

Evaluation Methods and Experiments

The most common approach in evaluating the accuracy of community mining algorithms is to report the result of the algorithm on well-known (typically small) real world datasets for which the ground truth is known like Zachary Karate Club data set [42, 22]. In this way, the accuracy is evaluated by comparing detected communities with the true communities in data. Another alternative is testing the algorithm on synthesized networks which are generated with characteristics similar to real networks and with a built-in community structure like the Girvan and Newman or LFR benchmarks [22, 35]. The scalability of the method could be further examined by applying the algorithm on large real networks for which there is no explicit notion of ground truth but we can check if the results are sound like with Amazon or DBLP datasets [12, 42]. For evaluating our proposed approach, we used all these three methods. Here we introduce our data sets and evaluation metrics, then report and discuss our results on them.

4.1 Data sets

4.1.1 Real Word Benchmarks

We have shown the accuracy of our approach by applying it over 6 well-known benchmark data sets.

Karate Club:

This network is drawn from the well-known study of Zachary [57]. In this study, relations between 34 members of a university karate club over a period of two years are observed and their network of friendships is constructed. The *Wkarate* dataset indicates the relative strength of these relations (number of times these people had interaction) while *Karate* dataset represents the presence or absence of interaction among the members. During the study, a disagreement developed between the administrator and the teacher of the club, which eventually made the club split into two smaller ones centering around the administrator and the teacher (represented by node 34 and node 1).

Strike:

This is the communication network of employees in a sawmill [45]. This data is collected in order to analyze the communication structure among the employees after a strike. Presence of an edge between two employees shows that they have discussed the strike with each other often. There are three groups as the ground truth according to age and language of the employees.

Football:

This dataset is the schedule for 787 games of the 2006 National Collegiate Athletic Association (NCAA) Football Bowl Subdivision [56]. In the NCAA network, there are 115 universities divided into 11 conferences. Additionally, there are 4 independent schools as well as 61 schools from lower divisions. Each school in a conference plays more often with schools in the same conference than schools outside. Independent schools do not belong to any conference and play with teams in all conferences, while lower division teams play very few games. The network contains 180 vertices (115 nodes as 11 communities, 4 hubs and 61 outliers), connected by 787 edges.

PolBooks:

This network represents books about US politics sold by the online bookseller Amazon [33]. It contains 105 nodes that represent books and 441 edges represent frequent co-purchasing of books by the same buyers (This is obtained from the feature of Amazon that indicates the ”customers who bought this book also bought these other books”). The ground truth illustrates whether these books are ”liberal”, ”neutral”, or ”conservative”.

PolBlogs:

This network represents the political leaning of blogs around the time of the 2004 presidential election[1]. It contains 1224 blogs from blog directories. 16715 links between blogs were automatically extracted from a crawl of the front page of the blog. The ground truth tells whether each blog is liberal or conservative.

4.1.2 Synthetic Networks

The most commonly used class of benchmarks to test community detection algorithms is presented by Girvan and Newman (GN) [22]. It contains 128 nodes and a built-in community structure with 4 groups of equal size. Lancichinetti et al. presented generalized benchmarks (LFR) which are more similar to real networks [35]. Unlike GN, in LFR benchmark, nodes can have different degrees and communities could be in varying sizes (derived from power law distributions) which is closer to heterogeneous distribution of nodes in real networks. In these benchmarks, each node shares a fraction of $1 - \mu$ of its edges with the other nodes of its community and a fraction of μ with the nodes of other communities, where $0 \leq \mu \leq 1$ is called the mixing parameter [20].

We have generated LFR benchmarks for networks of 5000 nodes with the average degree of 15, the maximum degree of 50 and different μ from $.1$ to $.9$. The community range is set from 200 to 500 nodes and the exponent of the degree distribution and community size distribution left as defaults (-2 and -1 respectively).

4.1.3 Large Scale Real Networks

This is a large network of Amazon.com, collected in August 2003 [12]. The nodes in the network are items such as books, CDs and DVDs sold on the website. Edges connect items that are frequently purchased together, as indicated by the "customers who bought this book also bought these items" feature on Amazon. There are 815,223 nodes and 3,426,127 undirected edges in this network.

4.2 Evaluation Metrics

We evaluated extracted communities by both comparing with ground truth and by measuring their modularity. Let V be the set of n nodes in the communities ($n = |V|$) and $R = R_1, R_2, \ldots, R_k$ denotes a partitioning (set of communities) on V such that $V = \bigcup_1^k R_i$ and $R_i \cap R_j = \phi$ for all $i \neq j$. We can evaluate R as follows:

4.2.1 Comparing with Ground Truth

With ground truth, validation is simply accomplished by means of comparison of communities, those discovered against the known communities. We used two measures of agreement between partitions typically employed for clustering evaluations: purity and Adjusted Rand Index. We compared R against partitioning G in the ground truth by computing these measures as follows:

Purity

is the number of correctly assigned nodes divided by the total number of nodes in V. Purity ranges from 0 (no agreement at all) to 1 (full agreement). It is computed using the following formula [37]:

$$purity(R, G) = \frac{1}{n} \times \sum_j max_i |R_j \cap G_i|$$

Adjusted Rand Index (ARI)

penalizes false negatives and false positives. ARI ranges between -1 (no agreement at all) and 1 (full agreement) with expected value of 0 for agreement no better than

random. Let a,b,c and d denote the number of pairs of nodes that are respectively in the same community in both G and R, in the same community in G but in different communities in R, in different communities in G but in the same community in R, and in different communities in both G and R. Given $R^n = \{R_i | n \in R_i\}$ and $\delta(R_i, R_j)$ is 1 if $R_i = R_j$ and 0 otherwise, the ARI is computed by the following formula [48]:

$$ARI = \frac{\binom{n}{2}(a+d) - [(a+b)(a+c)+(c+d)(b+d)]}{\binom{n}{2}^2 - [(a+b)(a+c)+(c+d)(b+d)]}$$

$$a = \sum_{ij} \delta(R^i, R^j)\delta(G^i, G^j)$$
$$b = \sum_{ij} (1 - \delta(R^i, R^j))\delta(G^i, G^j)$$
$$\dots$$

For data sets containing outliers or hubs, we considered seteach as two other communities.

4.2.2 Modularity

When ground truth is not available, modularity (Q) is typically used to assess the quality of discovered communities. It measures how well the edges fall within the detected communities compared to a randomized network. Let A be the adjacency matrix of the network containing m edges ($m = \frac{1}{2}\sum_{ij} A_{ij}$); then the portion of edges within communities is $\frac{1}{2m}\sum_{ij} A_{ij}\delta(R^i, R^j)$. The modularity is defined as substraction of this quantity from its expected value for a randomized network (with same nodes where edges created randomly but with respect the node degrees).

$$Q = \frac{1}{2m}\sum_{ij}[A_{ij} - \frac{d_i d_j}{2m}]\delta(R^i, R^j)$$
$$where \quad d_i = \sum_j A_{ij}$$

The modularity would be zero when the portion of within edge communities is no different than what we expect from a randomized network, and a value higher than 0.3 is a sign for significantly good partition [12]. For weighted networks, given the normalized weight matrix W, we generalize the definition of modularity

as follows:

$$Q = \frac{1}{2m}\sum_{ij}[W_{ij} - \frac{d_i d_j}{2m}]\delta(R^i, R^j)$$
$$where \quad d_i = \sum_j W_{ij}$$

4.3 Results and Discussions

In this section we report the results of our method and show a comparison with three of other well-known community detection methods; SCAN [56], CFinder [46] and FastModularity [12].

4.3.1 Comparing Initialization Methods

Table 4.1 shows the improvement of our results by developing the initialization of our algorithm. As shown in Table 4.1, even the Naïve initialization gives reasonable results but with high variance.

The Top Global Leaders improves the results significantly and reaches the maximum ARI in Karate and Strike but the best cases in the Naïve initialization for the Football data set still do better. This indicates that there is room for improvement. Examining the initial leaders obtained from the Top Global Leaders (TGL) and locating them in the network, indicates that some of these leaders are in the same community in the ground truth and choosing them as leaders, forces that community to split.

Top Leaders & not Direct Neighbour (TL&NDN) initialization do not improve the results which shows that the condition of not being direct neighbour is not a good one; since it would not avoid choosing leaders in the same community if they are not directly connected, which is very probable. It also may avoid choosing two true leaders which are in different communities and directly connected. The former is also very probable as the leaders are nodes with high centrality and may have links to outside of the community.

Top Leaders & not *iClose* (TL&NiC) method gives us the best result. This method makes a greedy selection, starting from the node with the highest centrality

method	dataset	ARI	purity	Q
Naïve	Karate	.80±.33	.90±.20	.28±.13
	Strike	.59±.25	.81±.13	.41±.12
	Football	.39±.12	.66±.08	.27±.07
TGL	Karate	1.0	1.0	0.37
	Strike	1.0	1.0	.54
	Football	.83	.88	.43
TL&NDN	Karate	1.0	1.0	0.37
	Strike	1.0	1.0	.54
	Football	.78	.88	.42
TL&NiC	Karate	1.0	1.0	0.37
	Strike	1.0	1.0	.54
	Football	.98	.97	.51

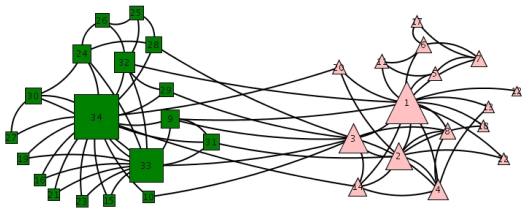
Table 4.1: Results of different initialization methods. For the Naïve method, average±standard deviation is calculated over 100 runs. All the results have the same default parameters (neighborhood threshold = 2, initialization threshold = 5 (nodes in common)) except the number of communities for each data set. (karate=2, strike=3, football=11) and the outlier threshold which is 4 for football but zero for karate and strike as we do not want to detect any outliers in those data sets.

to the lowest, we chose one if it does not have more than a threshold neighbours in common with any of the current leaders.

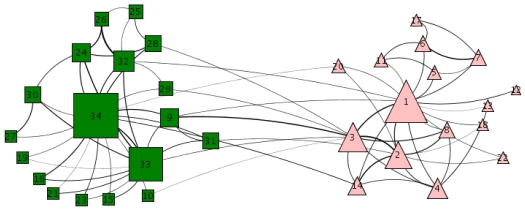
The visualized results of these three data sets are presented in Figure 4.1. These figures also show the correct communities as we obtained ARI 1 except for the football data set where we misidentified four hubs and assigned them to one community, obtaining an ARI of 0.98.

4.3.2 Comparing on real benchmarks

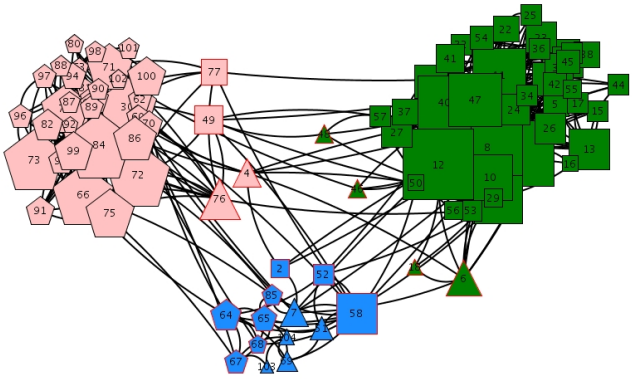
Table 4.2 shows a comparison between our approach (using Top Leaders & & not *iClose* initialization) and the three other algorithms on data sets described in section 4.1.1. Given the correct initial k, Top Leaders provides significantly better results. The other methods do not always find the correct k but even when that k seeded to Top Leaders, our approach improved the quality of the detected communities based on ARI. One interesting point in these results is the non-linear relation between modularity and ARI which suggest that optimizing the modularity would not necessarily increase the accuracy of the results and in most cases, the ground truth is



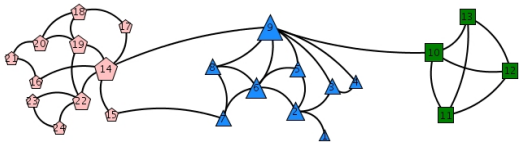
(a) Karate



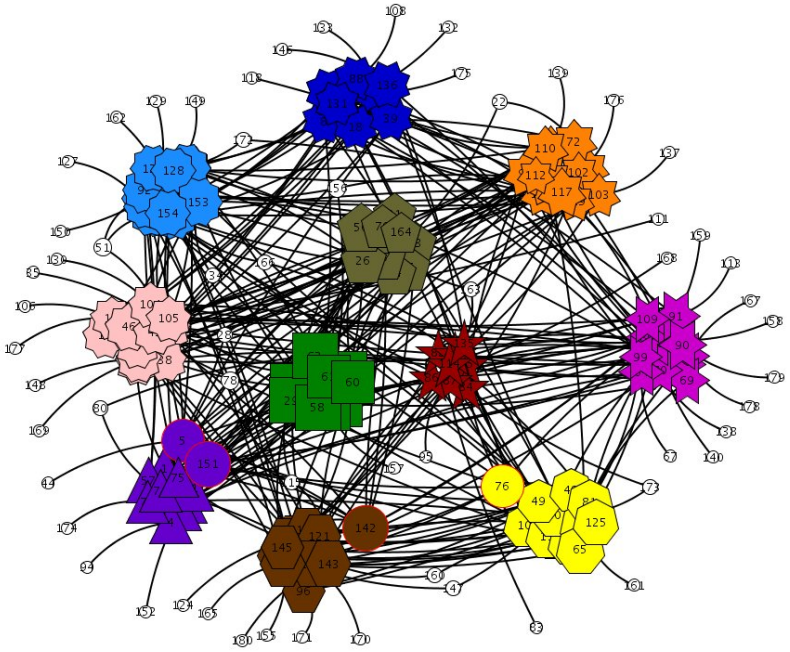
(b) WKarate



(c) PolBlooks



(d) Strike



(e) Football

Figure 4.1: Visualized communities detected using Top Leaders algorithm. Shape of a node represents its community in the ground truth, while the color of the node is according to its detected community by the Top Leaders. Moreover, size of nodes is based on their centrality in the network and thickness of edges is based on their weights.

closer to a partitioning with non-optimal modularity. Some of the visualized results of our approach given correct k can be seen in Figure 4.1.

4.3.3 Comparing on synthesized benchmarks

Figure 4.2 highlights the robustness of Top Leaders approach compared to FastModularity, CFinder and SCAN. Each plot compares the ARI of one of these methods and Top Leaders (seeded by the k suggested from the result of that method) on the networks with different mixing parameter described in 4.1.2. The corresponding results could also be seen in Table 4.3, where the average and variance of the plotted results (for synthesized networks with different mixing parameter μ) is reported.

4.3.4 Comparing on large scale data set

On the Amazon network, CFinder and SCAN did not terminate successfully; while Top Leaders obtained the result about 10 times faster than FastModularity (for the same $k = 2303$ obtained by FastModularity). The modularity of Top Leaders was 0.45 versus .77 of the FastModularity (which both guarantee that the detected communities are strong). However this could not show which algorithm is more accurate as the higher modularity does not ensure the higher accuracy (we have seen this in the previous experiments). Moreover our algorithm detected 89865 hubs which are not member of any specific community and this would decrease the modularity significantly (based on its definition presented in 4.2.2).

4.4 Parameters

Our main parameter is the number of communities in the network. This should either be given by domain experts or obtained from another algorithm for community mining that does not require k. However, even algorithms claiming not to require this parameter do not find the right number in many cases: FastModularity finds 12$\pm$6, CFinder finds 1182$\pm$464 and Scan finds 299$\pm$127 communities in synthesized benchmarks while the average size of communities in the ground truth is 33$\pm$5. Based on the results in Table 4.2 and Figure 4.2, using Top Leaders after an-

dataset	method	k	ARI	purity	Q
Karate (2 groups, 34 nodes, 78 edges)	fastModularity	3	.680	.970	**.380**
	CFinder	3	.705	.065	.182
	TopLeader(3)		**.838**	**1.0**	.374
	SCAN	4	.314	.764	.312
	TopLeader(4)		**.788**	**1.0**	.361
	TopLeader(2)		**1.0**	**1.0**	.371
WKarate (2 groups, 34 nodes, 78 edges)	fastModularity	3	.802	1.0	**.434**
	CFinder	3	.705	.065	.194
	TopLeader(3)		**.838**	**1.0**	.404
	SCAN	4	.319	.735	.339
	TopLeader(4)		**.665**	**1.0**	.416
	TopLeader(2)		**1.0**	**1.0**	.403
Strike (3 groups, 24 nodes, 38 edges)	fastModularity	4	.664	.958	**.555**
	TopLeader(4)		**.935**	**1.0**	.532
	CFinder	6	.348	1.0	.485
	TopLeader(6)		**.609**	**1.0**	.457
	SCAN	3	.848	.958	.547
	TopLeader(3)		**1.0**	**1.0**	0.548
PolBooks (3 groups, 105 nodes, 441 edges)	fastModularity	4	.637	.838	.501
	CFinder	4	.630	.814	.469
	SCAN	4	.599	.819	.499
	TopLeader(4)		**.649**	**.838**	**.517**
	TopLeader(3)		**.639**	**.828**	.498
Football (11 groups, 180 nodes, 787 edges)	fastModularity	7	.206	.427	**.567**
	TopLeader(7)		**.758**	**.777**	.445
	CFinder	12	.983	.913	.532
	TopLeader(12)		**.993**	**.977**	.511
	SCAN	11	1.0	1.0	.501
	TopLeader(11)		.988	.977	.513
PolBlogs (2 groups, 1224 nodes, 16715 edges)	fastModularity	12	**.892**	**.954**	**.311**
	TopLeader(12)		.835	.942	.283
	CFinder	-	-	-	-
	SCAN	74	.541	.407	.166
	TopLeader(74)		**.749**	**.949**	**.256**
	TopLeader(2)		.882	.939	.293

Table 4.2: Comparison of Top Leaders and other approaches on real benchmark datasets. Column k indicates the number of communities obtained by running the corresponding method.

method	k	ARI	purity
fastModularity	12.8 ± 6.8	.817 ± .098	.346 ± .259
+TopLeader		**.886** ± .042	.334 ± .262
CFinder	1182.6 ± 464.3	.884 ± .075	.713 ± .155
+TopLeader		**.949** ± .007	.788 ± .220
SCAN	299.7 ± 127.0	.670 ± .194	.374 ± .287
+TopLeader		**.948** ± .009	.673 ± .326
TopLeader(33.2±5.4)		.939 ± .021	.499 ± .297

Table 4.3: Comparing the accuracy of Top Leaders and other approaches on synthesized benchmarks: reported results are averaged for the synthesized networks with different networks mixing parameter μ.

other community detection approach would increase the quality of the final results significantly. In other words, if we trust the number of communities discovered by a community mining approach, seeding this number to Top Leaders would increase the quality of the discovered communities.

The other important parameter is the outlier threshold, γ. This parameter shows how *iClose* should a node be to a leader to be consider connected to that leader. That is, if a node has less than γ common friends with the leader, it does not have a considerable intersection with it and could not be part of its community. γ is an integer set to 0 (no outlier detection) or adjusted for the amount of noise to remove. For example the results reported in Table 4.2 for Football data sets obtained by setting γ to 4.

For the two other parameters, our algorithm does not exhibit any sensitivity and can remain at their default values. The first is the *iCloseness* threshold used in the initialization. It is a small constant showing the maximum number of common friends that two leaders in different communities can have. It is related to the average degree of nodes in the network and it is set to 5 in all our experiments, indicating that two leaders may be in the same community if they have more than 5 common friends. The second is the neighbourhood threshold, δ, which shows to what extent we should expand our neighborhood to find the winner leader. Setting this parameter to 2 gives us the reported results and increasing it does not change the results. This is due to the *small world phenomenon* in social networks i.e. the diameter of the network – longest shortest path – is increasing logarithmically with the size of

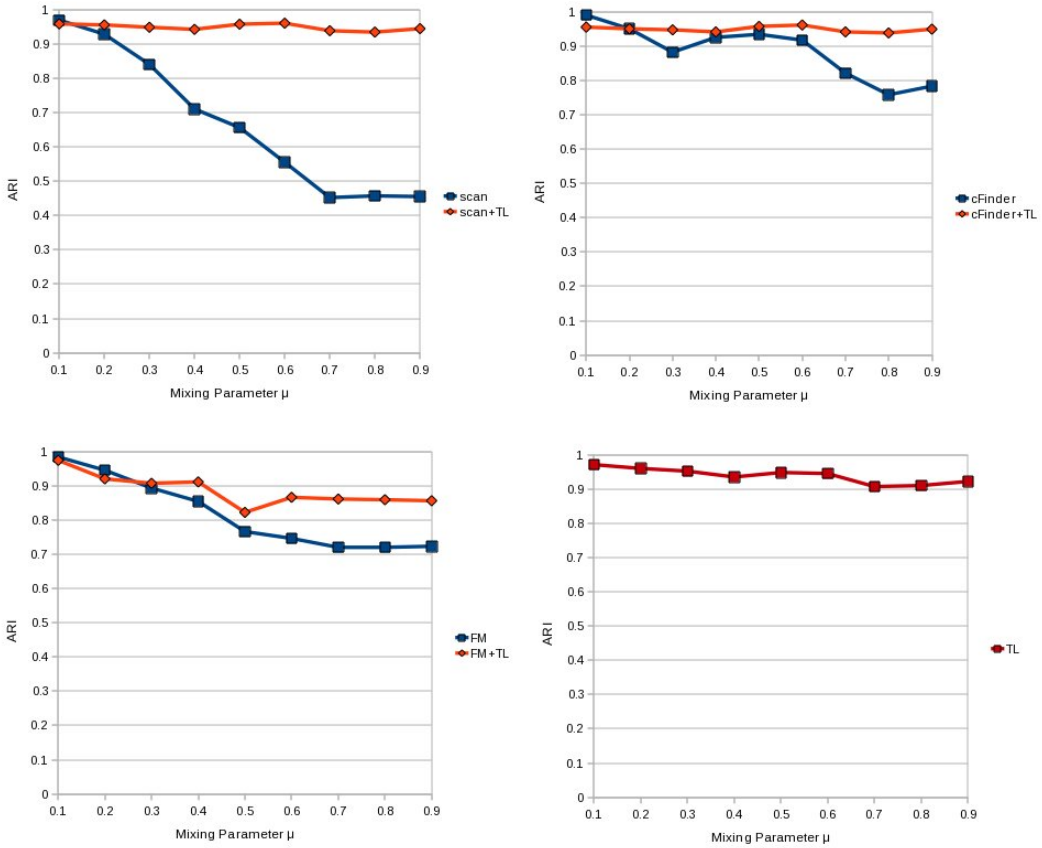


Figure 4.2: Comparison of Top Leaders and other approaches on LFR benchmark networks. Each plot compares one of the methods against Top Leaders which shows their obtained ARI as a function of mixing parameter μ. The Figure 4.2d shows the results of Top Leaders given the correct k.

the network which indicates that large social networks have small diameters [20].

4.5 Complexity

The complexity of our algorithm is $O(kn)$, the proof of which is similar to the one for k-medoids. If we fix the maximum number of iterations to some constant, in every iteration all the nodes should be assigned to one of the k leaders, which takes $O(kn)$ (the computation time of iCloseness could be neglected as it is only computing intersections of neighbourhood and is not a function of n or k). Based on our experiments, Top Leaders converges in a pretty small number of iterations. We have fixed the maximum number of iterations to 10 in all of our experiments

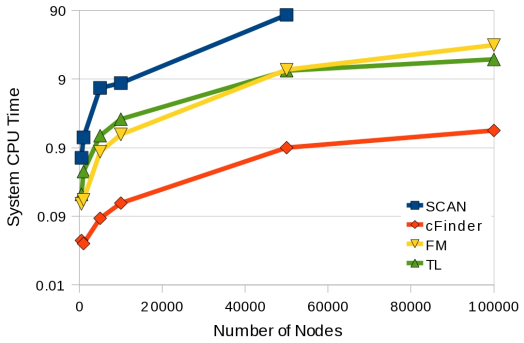


Figure 4.3: Comparing running time of Top Leaders and other approaches for different number of nodes in the network.

an none reached that, even the Amazon, our largest dataset, converged after only 6 iterations.

Besides, our implementation is much faster than $O(kn)$, as in every step, for each node, we only considered the leaders in its view, $\aleph(n, 2 \times \delta)$, whom are the leaders that might possibly have common neighbours of depth δ with that node.

Comparing to other methods, the time complexity of FastModularity is $O(nlog^2n)$ [12]. SCAN is reported in [56] to be in order of $O(m)$ where m is the number of edges in the network; however as it is expanding a community for each core node, it seems that the order of their algorithm is actually $O(nc)$, where c is the average size of the communities in the network. In Supplementary Information of [46], it is stated that finding full sets of cliques in a network is non-polynomial, while the efficiency of CFinder is shown by reporting its actual running time on large networks [46].

Here, we similarly show the running time of Top Leaders and compare it with FastModularity, CFinder and SCAN. Figure 4.3 illustrates this comparison. We have generated datasets with the same parameters as described in section 4.1.2 but for varying number of nodes ($\mu = .5$). Obviously all the experiments performed on the same machine (Quad-Core AMD Opteron Processor 8378). We reported the CPU time used by the program, which is used as a point of comparison for CPU usage of a program [1].

[1] `http://en.wikipedia.org/wiki/CPU_time`

Part II

Meerkat-ED: A Social Network Analysis Toolbox for Analyzing Participation of Students in Online Courses

Chapter 5

Social Network Analysis of Asynchronous Discussions in Online Courses

After introducing social network analysis and community mining approaches in the previous part, this part elaborates on an interesting application of those approaches in the context of on-line Education. Here we present how one could extract meaningful information about participation of students in online courses using social network analysis techniques, including community mining. In this chapter we first illustrate the place and need for social network analysis in studying the interaction of users in computer-supported collaborative learning environments. We continue by summarizing some recent studies in this area. In the next chapter, we present our specific toolbox for social network analysis of online courses, and illustrate its practical application on our own case study data.

5.1 Intoduction

There is a growing number of courses delivered using e-learning environments, especially in postsecondary education, using computer-supported collaborative learning (CSCL) tools, such as Moodle[1], WebCT[2], Blackboard[3], etc. Online asynchronous discussions in these environments play an important role in collaborative

[1] http://en.wikipedia.org/wiki/Moodle
[2] http://en.wikipedia.org/wiki/WebCT
[3] http://en.wikipedia.org/wiki/Blackboard_Learning_System

learning of students. It makes them actively engaged in sharing information and perspectives by interacting with other students [17].

In CSCL, there is a theoretical emphasize on the role of threaded discussion forums for online learning activities. Even basic CSCL tools enable the development of these threads where the learners could access text, revise it or reinterpret it; which allow them to connect, build, and refine ideas, along with stimulating deeper reflection [6].

Even in courses with a few number of students, there could be thousand of messages generated in a few months within these forums, containing long discussion threads bearing many interactions between students. Therefore the CSCL tools should provide a means to help instructors for evaluating participation of students and analyzing the structure of these interactions; which otherwise could be very time consuming for the instructors to be done manually.

Unfortunately, current CSCL tools do not provide much information regarding the participation of students and structure of interactions between them in discussion threads. In many cases, only some statistical information is provided such as frequency of postings, which is not a useful measure for interaction activity [17]. This means that the instructors who are using these tools, do not have access to a convenient indicators that would allow them to evaluate the participation and interaction in their classes [55]. Instructors usually have to monitor the discussion threads manually which is hard, time consuming, and prone to human error.

A large body of research exists on studying the participation of students in such discussion threads using traditional research methods: content analysis, interviews, survey, observations and questionnaires [15]. These methods try to detect the activities that students are involved in while ignoring the relations between students. For example, content analysis methods, as the most common traditional methods, provide deep information about specific participants. However, they neglect the relationships between the participants while their focus is on the content, not on the structure [55].

On the other hand, for fully understanding the participation of students, we need to understand their patterns of interactions and answer questions like who is

involved in each discussion, who is the active/peripheral participant in a discussion thread [15]. Nurmela et al. [44] demonstrated the practicality of social network analysis methods in CSCL, as a method for obtaining information about relations and fundamental structural patterns. Moreover, there is a recent line of work on applying social network analysis techniques for evaluating the participation of students in online courses like works done by Calvani et al. [6], Laat et al. [15], Willging et al. [55], Laghos et al. [34], and Erlin et al. [17].

The major challenges these works tried to tackle are: extracting social networks from asynchronous discussion forums (might require content analysis), finding appropriate indicators for evaluating participation (from Education's point of view) and measuring these indicators using social network analysis. None of them provides a complete or specific toolbox for analyzing discussion threads. However, they attempted to address one of these challenges to some extends. In the following, we present the related works on using social network analysis for evaluation of participation of students in online courses. First, we bring an overview of these methods and continue by describing each of these works in more detail.

5.2 Challenges: an Overview of Related Works

For applying social network analysis techniques to assess participation of students in an e-learning environment, we need to first extract the social network from the e-learning course, then consider which measures show an effective participation and finally report these measures in an appropriate way. Here, we bring an overview of the previous works related to each of these three phases.

5.2.1 Extraction of Social Network

CSCL tools record log files that contain the detailed actions that occur within them and hence information about the activity of the participants in the discussion forums [44]. Laat et al. [15], Willging et al. [55], Erlin et al. [17] and Laghos et al. [34] used these log files generated by the environment in which the course is held to extract the social network underneath of discussion threads. Laghos et al. stated

that they considered each message as directed to all participants in that discussion thread while others consider it as only directed to previous message.

Gruzd et al., [26] and [27], proposed an alternative and more complicated way of extracting social networks, called named network. They argue that using this common method (connecting a poster to the previous poster in the thread) would result in loosing much of the connections. Their approach briefly is: first using named entity recognition to find the nodes of the network, then counting the number of times that each name is mentioned in posts by others, to obtain the ties and finally weighting these ties by the amount of information exchanged in the posts. However, their final reported results are not that much promising and even obtaining those results requires many manual corrections during the process.

Regarding what we should consider as the participation in extracting the social network, Hrastinski [28] suggested that apart from writing, there are other indicators of participation like accessing the e-learning environment, reading posts or the quantity and quality of the writing. However, all of these methods extracted networks just based on posts by student (writing level).

5.2.2 Measuring the Effectiveness of Participation

In education context, Calvani et.al. [6] defined 9 indicators for measuring the effectiveness of participation to compare different groups within a class; extent of participation (number of messages), proposing attitude (number of messages with proposal label), equal participation (variance of messages for users), extent of role (portion of roles used), rhythm (variance of daily messages per day), reciprocal reading (portion of messages that have been read), depth (average response depth), reactivity to proposal (number of direct answers to messages with proposal label) and conclusiveness (number of messages with conclusion label).

Daradoumis et al. [13] defined high level weighted (showing the importance) indicators to represent collaboration learning process; task performance, group functioning, social support, and help services. They further divided these indicators to skills and sub-skills, and assigned every sub-skill to an action. For example, group functioning is divided into these skills: active participation behavior, task

processing, communication processing, etc. Where communication processing is itself divided into more sub-skills: clarification, evaluation, illustration, etc. and clarification is mapped to the action of changing description of a document or url.

However, for measuring the effectiveness of participation, most of the previous works simply used general social network measures (different centrality measures, betweenness, etc.), available in one of the common general social network analysis toolboxes. Laat et al. [15], Willging et al. [55], Erlin et al. [17] used UCINET[4] and Laghos et al. [34] used NetMiner[5].

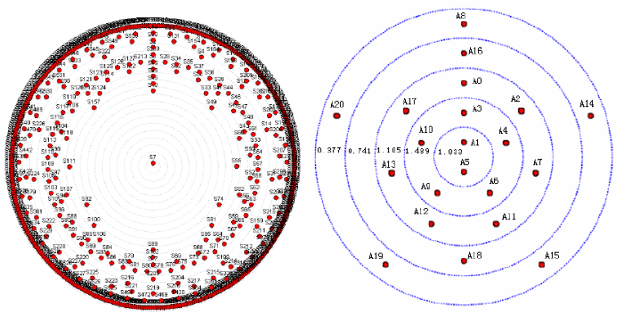
5.2.3 Results Representation

For results representation, Laat et al.[15] and Erlin et al. [17] only reported the plain results. Then they brought detailed discussions on these results and showed their soundness by specific case by case examples. Laghos et.al. [34] (Figure 5.1a) and Willging et al. [55] (Figure 5.1b) visualized the circular centrality graph which placed the more central/powerful nodes closer to the center. Willging further used MAGE toolbox to build a 3D graph of interactions where central students are placed in the center of the graph (Figure 5.3). Laghos et al. [34] plotted the structural profile sociogram which puts students that have similar interaction pattern closer to each other and is used to cluster students according to their level of activity (Figure 5.4). And Calvani et al. [6] drew a nonagon graph of their indicators which shows the group interactions relatively to the mean behavior of all groups (Figure 5.2).

5.3 Elaborate Description of Previous Attempts

Hrastinski [28] proposed a comprehensive definition for online learner participation by reviewing how other researchers interpreted it. He found out that participation is more than simplistic measures like the total number of student posting in a discussion forum or length of the discussion threads. He pointed out that online participation is defined in six different levels: accessing e-learning environment, writing, quality writing, writing and reading, actual and perceived writing, and taking part

[4] http://www.analytictech.com/ucinet/
[5] http://www.netminer.com/NetMiner/



(a) Figure reprinted from [34](b) Figure reprinted from [55]

Figure 5.1: Comparing Centrality of Students: the circular centrality graph which placed the more central/powerful nodes closer to the center.

and joining in a dialogue. Further he summarized quantitative and qualitative approaches for studying online learner participation; Quantity of messages (or units: word, phrases, sentences, thoughts, ideas), Message quality (categorized messages according to a classification scheme: on-topic/off-topic, asking/answering, etc.), Learner perceptions (included interviews, reflective learner reports and surveys), Message lengths, System accesses or logins, Read messages, Time spent.

To help a tutor in monitoring these groups, Calvani et al. [6] proposed 9 educational indicators for comparing several groups based on effectiveness of their interactions: extent of participation (number of messages), proposing attitude (bring forward ideas and proposals), equal participation(deviation standard (σ) of messages for users), etc. They analyzed threaded web forums within an add-on module for the Moodle (Forum Plus). This module added a label to each message which shows its type; whether it is a proposal of a new idea or discussion or summary, etc. It also logged the user's behavior like reading a message. Using this label, they computed their indicators which are all basically counting.

Willging et al. [55], stated that not only analyzing online interactions by content analysis or thread analysis is time consuming; but also it ignores structural characteristics of the interactions. They suggested that adding a tool based on SNA and visualization techniques to Learning Management Systems (LMS) could be helpful for instructors in better monitoring and assessing participation of students. They supported this suggestion by analyzing data of an online course held on Blackboard

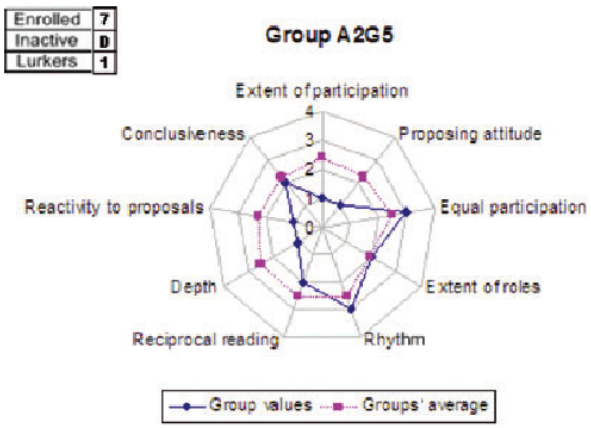


Figure 5.2: Comparing Participation of a Group: this nanogram illustrates a comparison of participation of one group (blue lines) with the average participation of other groups (red lines) using the nine indicators defined by Calvani in [6]. This figure is reprinted from [6].

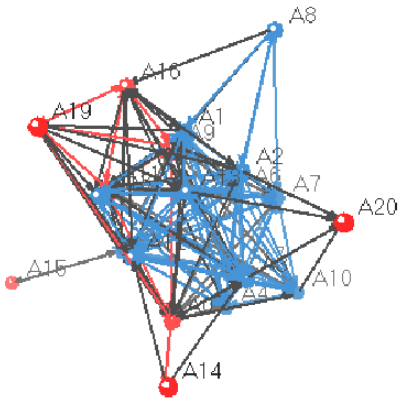


Figure 5.3: 3D Graph of Interactions: central students are placed in the center of the graph. This figure is reprinted from [55].

and consisting of 21 students. They used UCINET and computed in-degree, out-degree centrality and betweenness; they drew the circular centrality degree graph. They also visualized interactions of students and their interactions where more central students placed in the center of the graph.

Laat et al. [15] performed Social Network Analysis (SNA) on a course held on WebCT and consisted of seven students. They wanted to see how dense the interactions are and who are central participants. They used log files to build a case-by-vase matrix and further they used UCINET to perform SNA (computing out/in degree centralities) and drawing the sociogram (student are nodes and numbered

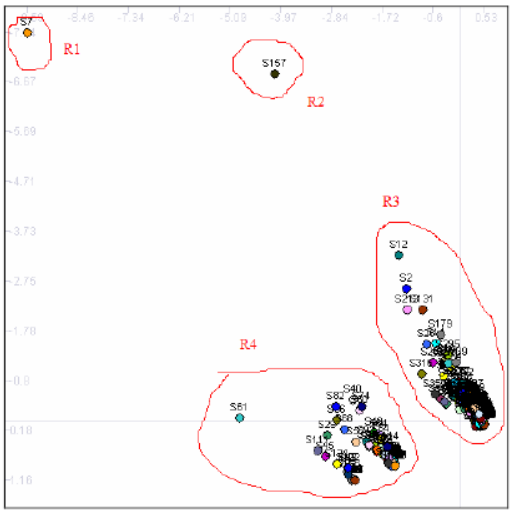


Figure 5.4: Structural Profile Sociogram: students that have similar interaction pattern placed closer to each other. This figure is reprinted from [34].

ties shows the volume of communications). They had analyzed this data before using Content Analysis (CA) and Critical Event Recall (CER) reviews; here they showed that these new outcomes agreed with their previous results using case by case examples.

Erlin et al. [17] used the data of transcripts of a discussion thread (of 12 students) generated by moodle (from which one could gather information about amount of messages, how many times and by whom a certain message was read). They treated this data as relational data, stored it in a matrix and analyzed its centrality, betweenness and closeness using UCINET and further drew the out-degree centrality graph.

Laghos et al. [34] used NetMiner SNA toolbox for analyzing the SN extracted from discussions, logged in an online self-taught course using moodle and with 128 students. They assumed each message is directed to all participants in the discussion thread. They reported results for connection (mean ego-network size, betweenness, link connectivity, bridges, cutpoints), cohesion (number of cliques with size k), centrality (mean, sd, min and max + mean out-degree sociogram: farther from the center, lower power in the network), and equivalence (mean of structural, regular and automorphic equivalence + structural profile sociogram: where actors with similar patterns of communication are closer to each other)

While all other previous methods used log files for extracting the social networks, Gruzd et al. [26], [27] proposed NameNetwork, an alternative content based method for building the social networks from discussion threads. In this method, first names are discovered by a complicated named entities recognition approach which also needs a manual checking at the end for obtaining acceptable accuracy. Then ties are extracted: two names are considered connected if the number of times they are mentioned in postings of each others is higher than a threshold. Finally, weights are assigned to these ties based on the amount of information exchanged in the postings; which is computed by counting the number of concepts (obtained using Yahoo! term extractor) in the posts divided by the length of the posts.

They referred to previous methods of extracting social networks out of posting headers as chain networks. In [25] three different methods for building the chain networks is considered: connecting a poster to the last person in the post, connecting a poster to the last and first (=thread starter)person in the chain and also connecting a poster to all people in the reference chain with decreasing weight. In [26] they stated the first chain network is more logical and practical comparing to other two. Comparing to NameNetwork, they claimed that about 40% of connections would be missed in the chain network (including connections mentioned in the first posting of the thread, connections that have one end outside of the current existing thread, connections where the addressee is not the most recent one in the thread). However (and with much of manual adjusting), the final extracted network is slightly better than chain network in 4 of their datasets but worse in two others.

Daradoumis et al. [13] have analyzed collaboration of students in an on-line course held on BSCW; where every 5-6 students formed a group (there are about 90 of these groups) to deliver a final project and they interacted both inside groups in a private space and also publicly in a general workspace. Their dataset contained log files of the actions performed by the group participants (create, change, read or move an object) and self-assessment reports.

They defined four high level weighted (showing the importance) indicators to represent collaboration learning process, which are assessed by the tutor frequently during the course (1. task performance: learning outcome, 2. group functioning

(interaction behavior), 3. social support, 4. help services (task scaffolding)). They further divided the first two indicators to skills and sub-skills, and assigned every sub-skill to an action. For example, group functioning is divided into 5 skills: 1. active participation behavior, 2. social grouping, 3. task processing, 4. workspace processing, and 5. communication processing. Where communication processing is divided into 5 sub-skills: 1. clarification, 2. evaluation, 3. illustration, etc. and clarification is mapped to these actions: change description/ change event doc, change description url.

They analyzed the first two skills in group functioning by SNA tools named SAMSA which makes the sociomatrix based on desired date, actors and relationship type. They considered relationships between every actor that creates an object in BSCW workspace and those that access this object in order to read it. They have computed network density, degree centrality, degree centralization (group-level measure based on degree centralities of actors).

In this chapter we illustrated the applicability of social network analysis in the study of students' interaction in e-learning environments. We summarized the recent studies in this area which do not address the problem completely but tackle with some of its challenges to some extends. In the following chapter, we proposed our specific toolbox for analyzing students interactions in asynchronous discussion forums.

Chapter 6

Meerkat-ED: Social Network Analysis toolbox for Education

In this chapter we present our specific toolbox, named Meerkat-ED, for social network analysis of online courses, and illustrate its practical application on our own case study data. Meerkat-ED is a specific social network analysis toolbox for visualizing, monitoring and evaluating participation of students in discussion forums of online courses.

6.1 Introduction

Meerkat-ED helps instructors in assessing the participation of students in asynchronous discussion forums of online courses. It analyzes the structure of interactions between students in these discussions using social network analysis techniques. It prepares and visualizes overall snapshots of participants in the discussion forums, their interactions, and the leaders/peripheral students in these discussions.

Moreover, It creates a hierarchical summarization of the topics discussed in the forums using community mining, which gives the instructor a quick view of what is under discussion in these forums. It further illustrate how much each student has participated on these topics, by showing his/her centrality in the discussions on that topic, the number of posts, replies, and the portion of terms used by that student in discussions on that topic.

Meerkat-ED builds and analyzes two kinds of networks out of the discussion forums: social network of the students (links represent correspondence) and network

of the phrases used in the discussions (links represent co-occurrence of phrases in the same sentence). Interpreting the first network shows the interaction structure of the students participated in the discussions. Moreover, centrality of students in this network, lay out a notion of their leadership in the course.

Interpreting terms network, depicts the terms used in the discussion and the relations between these terms. Finding the hierarchical communities in this network, demonstrates the topics addressed in the discussions. While choosing each of these topics will outline the students who participated in that topic and the extent of their participation.

In the rest of this chapter, we first describe our case study data set which is used for showing the practicability of Meerkat-ED. Then we show how we extract, analyze and interpret the social network of the students and terms.

6.2 Practical Application

The data set we have used is obtained from a postsecondary course. The course is titled Electronic Health Record and Data Analysis (CMPUT 690) and was offered in Winter 2010 at University of Alberta. The permission to use the anonymized course data for research purposes was obtained from all the students registered in the course.

This data is further anonymized by assigning fake names to students and replacing any occurrence of first, last or username of the students in the data (including content of the messages in discussion forums) with the assigned fake name. We also removed all email addresses from the data.

In this course, as it is also usual in other courses, the instructor initiated different discussion threads. For each thread he posted a question or provided some information and asked students to discuss the issue. Consequently students posted subsequent messages in the thread, responding to the original question or to the response of other students.

This course was offered using Moodle which is a widely used course management system. Moodle like other CSCL tools, enables interaction and collaborative

construction of content, mostly using its Forum tool which is a place for students to share their ideas [1]. Only using Moodle, to evaluate student participation the instructor is limited to shallow means such as the number of posts per thread and eventually the apparent size of messages. He would have to manually monitor the content of each interaction to measure the extent of individual participation, which is hard, time consuming and even unrealistic in large classes or forums with large volume.

Moodle also provides a functionality to record all the information and resources of a course in a backup xml file. We have parsed this backup file to extract the course information including the characteristics of students (firstname, lastname, username, email address, etc.) and information regarding the discussion threads (sequence of details about messages in the different threads: title, content, date, author, parent message, etc.).

We have further used Meerkat-ED to build and analyze two kind of networks from these information: the social network of students and the network of the terms used by them. The instructor of the course denoted the usefulness of the results of these analysis in evaluating the participation of students in the course. In this experiment, the instructor reported that using MeerkatED it was easy to have an overview of the whole participation and it was possible to identify influential students in each thread as well as identify quiet students or unvoiced opinions, something that would have been impossible with the simple statistics provided by Moodle.

6.3 Interpreting Students Interaction Network

Interpreting the network of interaction between students, helps instructors monitor the interaction structure of students, and examine which students are the leaders in given discussions and who are the peripheral students. Here we first describe how the network is built based on the information we have from discussion threads. Then we visualize the extracted network and continue by bringing an analysis of leadership of the students based on their centrality in this network.

[1] `http://moodle.org/about/`

6.3.1 Student Network Extraction

The student network shows the interaction between students in the discussion forums. In this network, the nodes represent students of the course and edges are the interaction between these students. An interaction between two students is weighted by the number of messages passed between them.

This network could be built both directed or undirected (which is chosen by the instructor); where in the directed model, each message in considered connecting the author of the message to the author of its parent message.

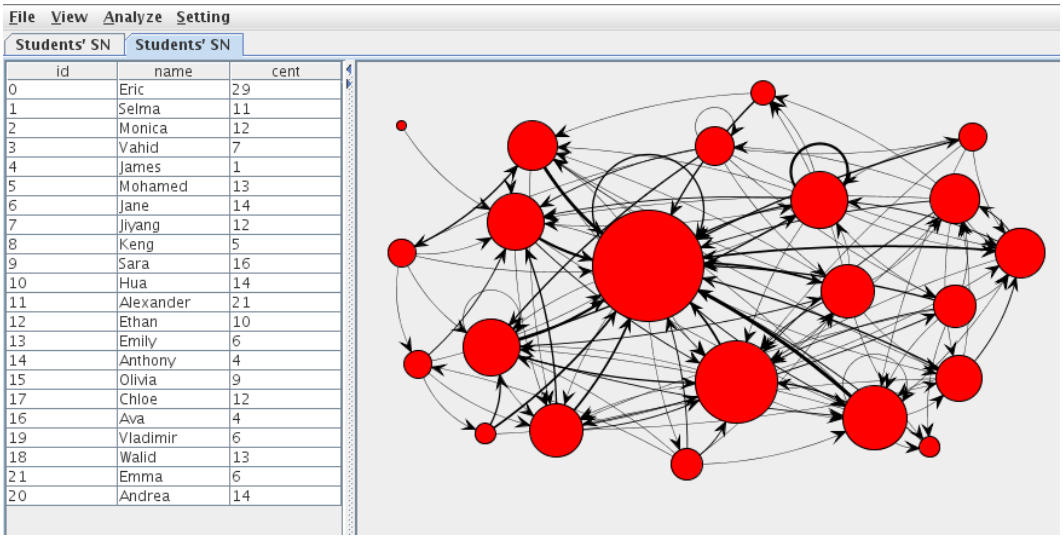
6.3.2 Visualization of Student Network

Figure 6.1 shows the visualized network of students in the course. The size of the nodes corresponds to their degree centrality in the network. This means that the bigger a node is, the more messages the student represented by that node sent and received.

The thickness of the edges in the network represents the weight of interactions which is based on the number of messages in the interaction of communicating students. Choosing an edge would bring up a pop up window that shows these messages as illustrated in Figure 6.2.

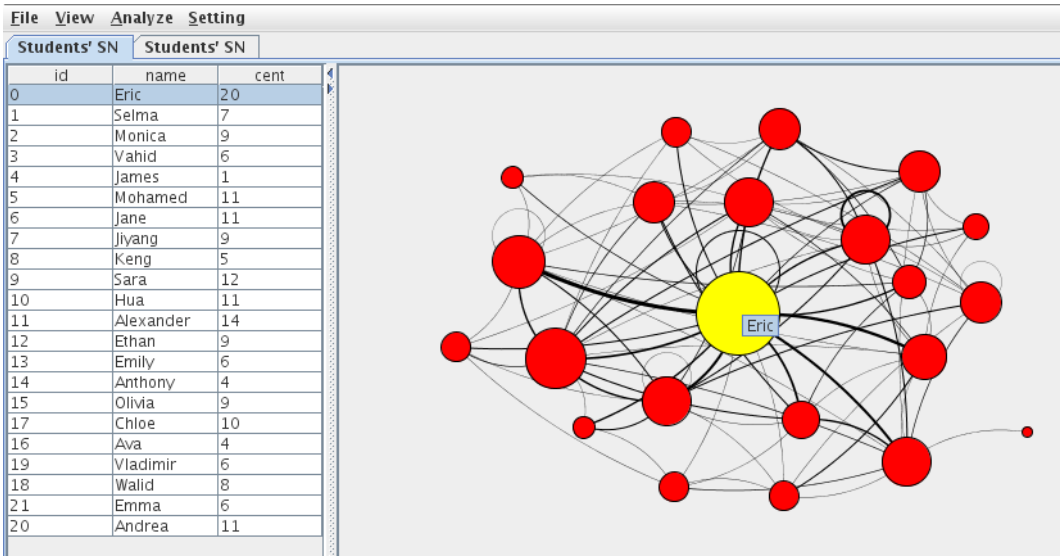
6.3.3 Analyzing Leadership in the Student Network

The leadership and influence of students in the discussions could be compared by examining the centrality of nodes corresponding to them in the network; as the nodes' centrality measures their relative importance within a network. The nodes' centrality is depicted by the size of the nodes in the visualized network as illustrated in Figure 6.1. Moreover, students could be ranked more explicitly in a circular centrality graph in which the more central/powerful the node is, the closer it is to the center, as presented in Figure 6.3.



id	name	cent
0	Eric	29
1	Selma	11
2	Monica	12
3	Vahid	7
4	James	1
5	Mohamed	13
6	Jane	14
7	Jiyang	12
8	Keng	5
9	Sara	16
10	Hua	14
11	Alexander	21
12	Ethan	10
13	Emily	6
14	Anthony	4
15	Olivia	9
17	Chloe	12
16	Ava	4
19	Vladimir	6
18	Walid	13
21	Emma	6
20	Andrea	14

(a) Directed Network



id	name	cent
0	Eric	20
1	Selma	7
2	Monica	9
3	Vahid	6
4	James	1
5	Mohamed	11
6	Jane	11
7	Jiyang	9
8	Keng	5
9	Sara	12
10	Hua	11
11	Alexander	14
12	Ethan	9
13	Emily	6
14	Anthony	4
15	Olivia	9
17	Chloe	10
16	Ava	4
19	Vladimir	6
18	Walid	8
21	Emma	6
20	Andrea	11

(b) Undirected Network

Figure 6.1: Visualized Student Network: The left panel lists the students in the course. The right panel shows the social network of interaction of students in the course. The size of nodes corresponds to their centrality/leadership in the discussions. The width of edges represents the weight of communication between incident nodes.

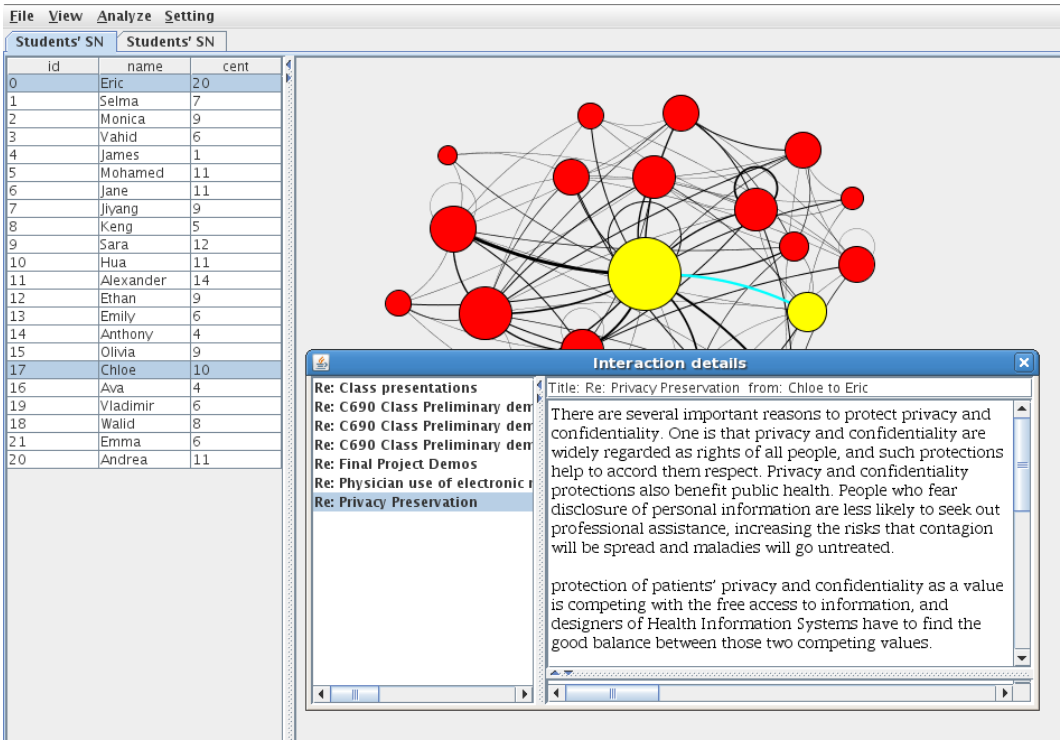


Figure 6.2: Visualization of messages in an interaction: the interaction window shows the messages passed between nodes incident to the selected edge: Chole and Eric. Selecting each message from the left panel would show its title, sender, receiver and content.

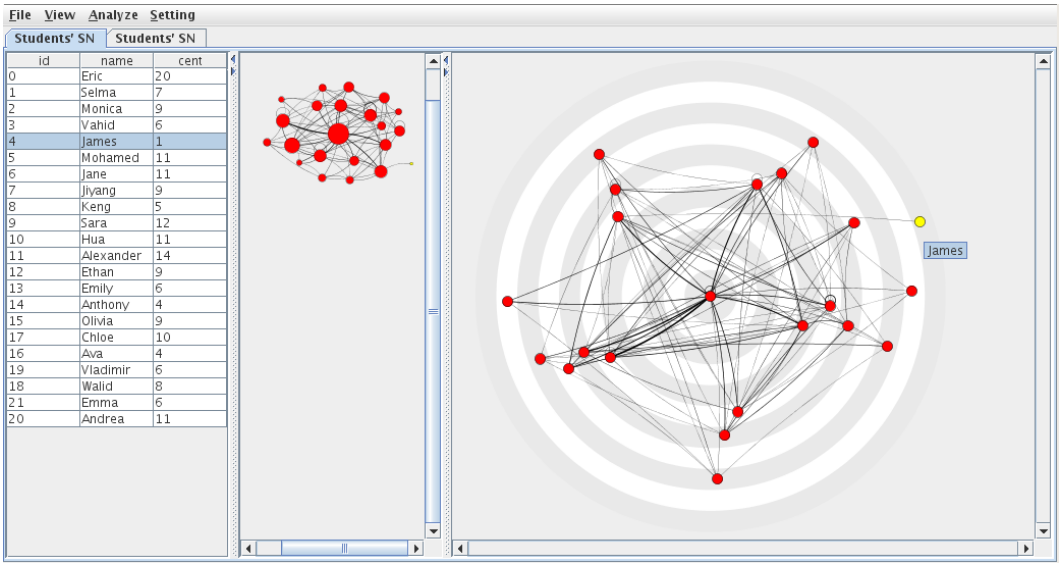


Figure 6.3: Comparing centrality of students: the students closer to the center are more central in the student network, i.e., have participated more in the discussions of the course. Likewise, the further from the center, the less the student was active; here James is the least active students in the discussions and is placed on the outer circle.

6.4 Interpreting Term Network

Interpreting the term network, depicts the terms used in the discussions and the relation between these terms. Moreover, finding the hierarchical communities in this network, demonstrates the topics discussed in the discussions. While choosing each of these topics will outline the students who participated in that topic and the extent of their participation. In the following, we first describe how the network is extracted from the discussions, then we show example of obtained network visualization and present how it could be interpreted for evaluation of participation. Finally, we show the topics (term communities) and their explication.

6.4.1 Term Network Extraction

In the term network, nodes represent noun phrases occurred in the discussions; and edges show the co-occurrence of these terms in the same sentence. Each co-occurrence edge contains the messages in which its incident terms occurred together; and is weighted by the number of sentences these terms co-occurred.

For building this network, we need to first extract the noun phrases from the discussions, then build the network by setting the extracted phrases as nodes and checking their co-occurrence in all the sentences of every message for creating the edges.

Extracting Terms from Forums

We have used the OpenNlp toolbox for extracting noun phrases out of discussions. OpenNlp is a set of natural language processing tools for performing sentence detection, tokenization, pos-tagging, chunking, parsing, and etc.[2]

Using sentence detector in OpenNlp, we first segmented the content of messages to their consisting sentences. The tokenizer was used to break down those sentences to words. Having the tokenized words, we used the pos-tagger to determine their part of speech, whether they are noun, verbs, adjective, etc. Then using the chunker, we grouped these words to the phrases, and we picked the detected noun phrases,

[2] `http://opennlp.sourceforge.net/README.html`

which are sequences of words surrounding at least one noun and functioning as a single unit in the syntax.

Pruning

For obtaining better sets of terms to represent the content of the discussions, pruning on the extracted noun phrases was necessary. We removed all the stopwords, and split the phrases that have stop word(s) within into two different phrases. For example the phrase ”privacy and confidentiality” is split into two terms: “privacy”, and “confidentiality”.

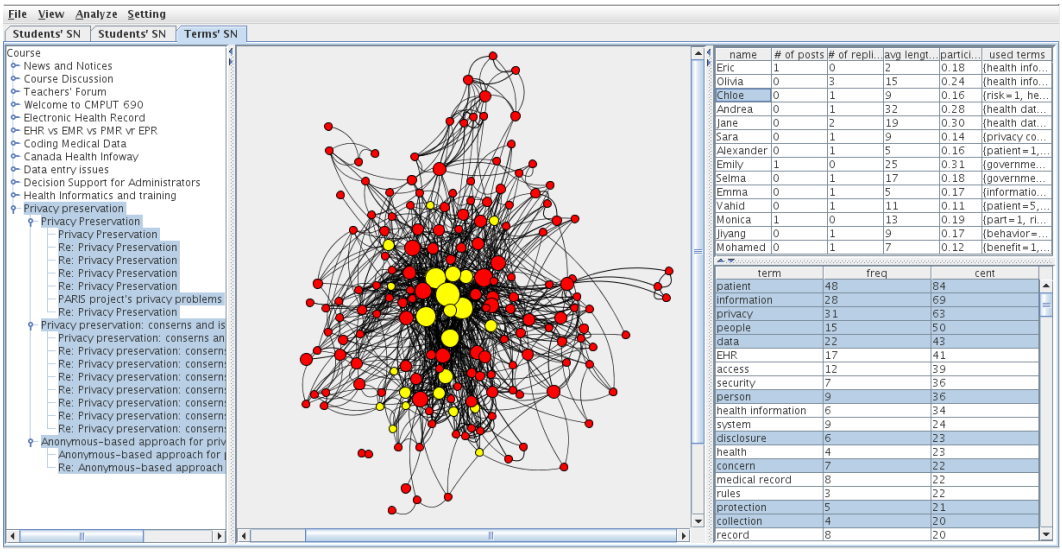
To avoid having duplicates, the first characters were converted to lower case (if the other characters of the phrase are in lowercase) and plurals to singular forms (if the singular form appeared in the content). For instance “Patients” would be “patients” then “patient”.

As the final modification, we have applied a common filter, which is removing all the noun phrases that just occurred once; which would prune most of unwanted phrases.

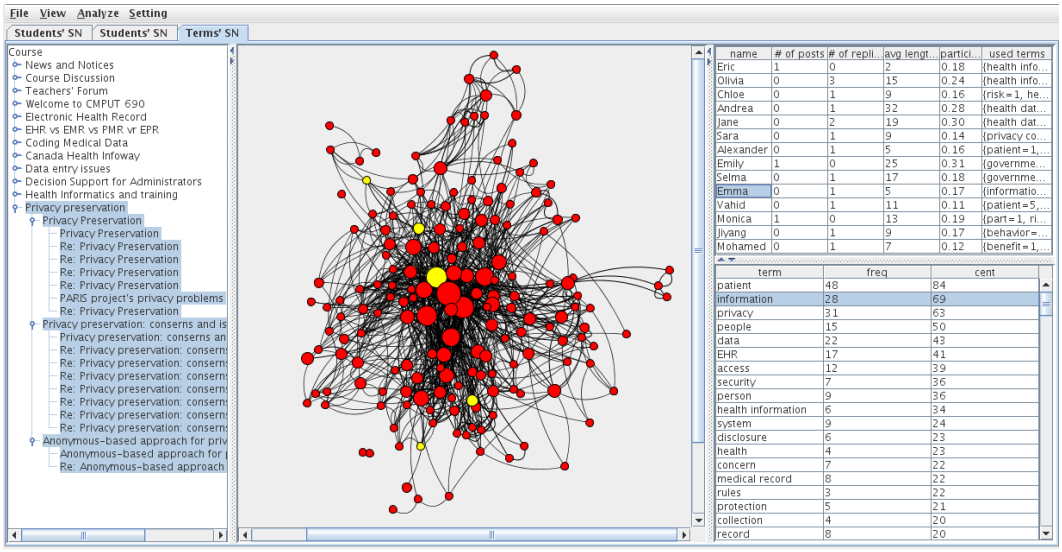
6.4.2 Visualization of Term Network

Figure 6.4 presents the visualization of the term network. In this figure, the size of the nodes represents the frequency of their corresponding terms and the thickness of edges represents the weight of the co-occurrences, which is the number of sentences in which incident terms occurred together. Selecting an edge would show these messages as illustrated in Figure 6.5.

In this visualization the instructor would see a list of the discussion threads in the course while selecting any set of those discussions/messages would bring up the corresponding term network, along with the list of terms occurring in them and the list of students that participated in these selected set of discussions/messages. Selecting any of these terms would show the students that used that term. Likewise, selecting any of the students would outline the terms used by the student, as illustrated in Figure 6.4a and 6.4b; which is highlighting the differences between participation of the students.



(a) Terms used by Chole



(b) Terms used by Emma

Figure 6.4: Visualized Term Network: The left panel lists the discussion threads in the course. The middle panel shows the network of terms in the selected set of discussions. The upper right panel shows list of students participated in the selected discussions, along with some statistics about their participation such as number of posts, replies, etc. The bottom right panel shows the terms used in these discussions. Selecting each student, would outline the terms used by that student.

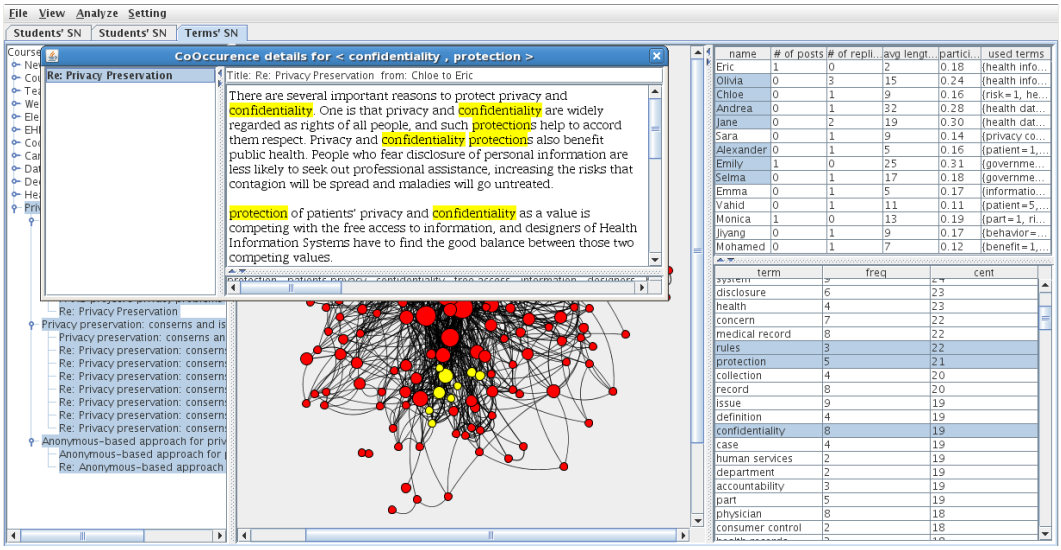


Figure 6.5: Co-occurrence of terms: selecting a co-occurrence edge would bring up a pop op window that shows the messages these incident terms co-occurred together in, highlighting the corresponding terms in the content.

6.4.3 Finding Term Communities (Topics)

The term Network could be further analyzed to group the terms co-occurring mostly together. These groups represent the different topics discussed in the messages and could be obtained by detecting the communities in the term network.

For creating the hierarchy of the topics, we applied a community mining algorithm repeatedly to divide one of the current connected components of the network, until the size of all components is smaller than a threshold, $\alpha = 5$, or the division of any of the components would result in a partitioning with modularity less than a threshold, $\beta = 0.1$. We used FastModularity [12] as the community detection algorithm, however it could be any other community mining approach such as Top Leaders, presented in the first part of this dissertation.

Figure 6.6 shows the detected topics (term communities) in the network given in Figure 6.4. The green nodes show the representative nodes of communities. Each representative node, contains 10 most central terms of the terms in the community it represents. The size of the representative nodes corresponds to the number of terms in their communities; while the size of the leaf nodes, terms, is related to their frequency, same as the term network. Similar to the term network, here also

one could select a set of terms, usually within a topic, to see who participated in a discussion with that topic and to what extent, as illustrated in Figure 6.7.

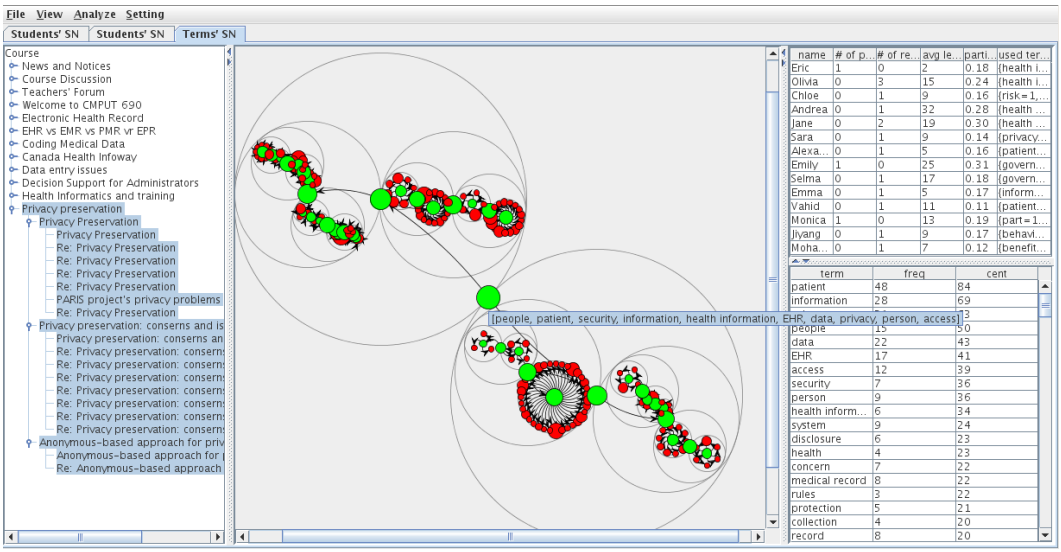


Figure 6.6: Term communities (Topics): The gray circles outline the communities boundaries and the green nodes represent the community representatives. Each community representative is accompanied with its top 10 phrases in its community. These could be seen in the tooltip in the figure.

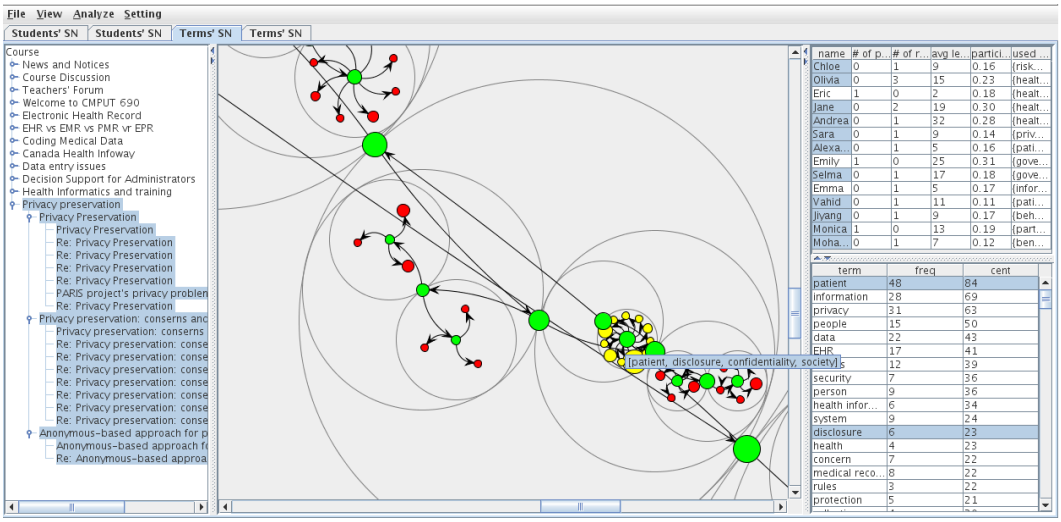
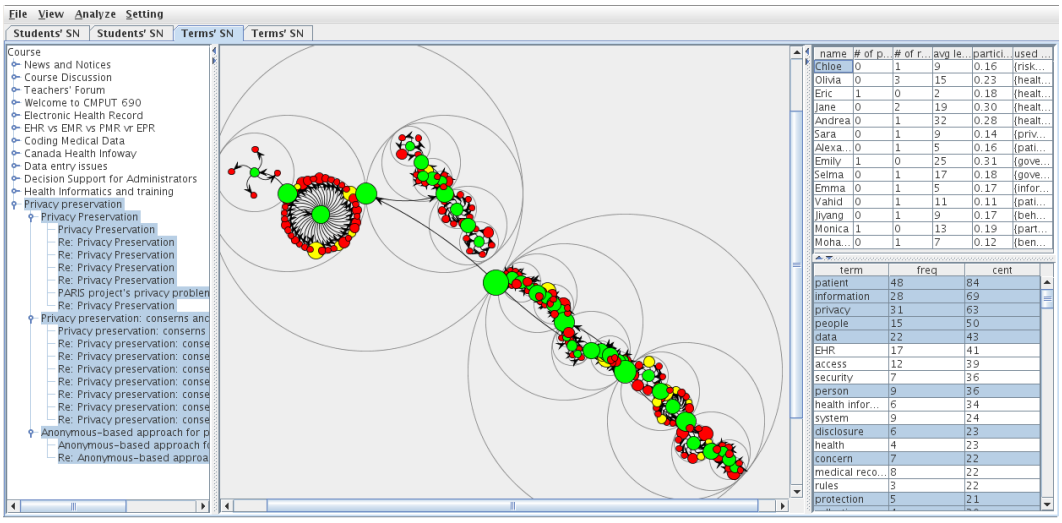
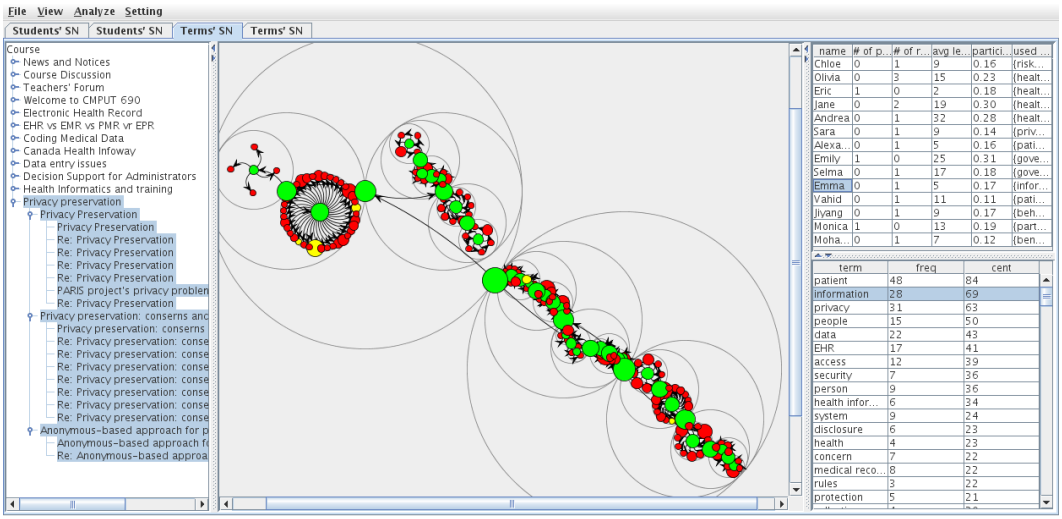


Figure 6.7: Term communities (Topics), zoomed: selecting each topic, would outline the students who participated in a discussion with the topic, and the terms in that topic. Here, the topic is roughly about ”patient, disclosure, confidentiality and society”. Moreover, students who participated in this topic and their contribution could be seen in the upper right panel.



(a) Terms used by Chole



(b) Terms used by Emma

Figure 6.8: Comparing participation range in the topics: The left panel lists the discussions threads in the course. The middle panel shows the topics discussed in the selected set of discussions. The upper right panel shows list of students participated in the selected topics, along with some statistics about their participation. Here we can see that Chole had a wider participation in this discussion thread, comparing to Emma as she participated in more topics.

For evaluating the participation of students, one might check how wide were their participation. In other words, students who participated in different topics could be considered more active than students that just talked about a smaller number of topics. This could be examined by selecting each student and checking how many topics he/she participated in as illustrated in Figure 6.8. In this chapter, we

then proposed Meerkat-ED, a specific and practical toolbox for analyzing students interactions in asynchronous discussion forums. Our toolbox prepares and visualizes overall snapshots of participants in the discussion forums, their interactions, and the leaders/peripheral students. Moreover, It creates a hierarchical summarization of the discussed topics, which gives the instructor a quick view of what is under discussion. It further illustrates individual student participation in these topics, measured by their centrality in the discussions on that topic, their number of posts, replies, and the portion of terms used by them. We believe exploiting the mining abilities of this toolbox would facilitate fair evaluation of students' participation in online courses.

Chapter 7

Conclusion

7.1 Conclusions

In this dissertation we elaborated the importance of social network analysis for mining structural data and its applicability in the domain of education. In summary:

1. In Chapter 2 we introduced social network analysis and community mining for studying the structure in relational data. We surveyed the traditional approaches and further elaborated on recent methods which borrow different concepts from social network analysis to investigate the community structure. Along with Chapter 3, this chapter addressed the first statement of the thesis.
2. We addressed our second statement in Chapter 3, where we established a closeness measure, which we called Intersection Closeness, to assess the proximity of a node to a community representative. This measure *iCloseness* is based on the theory of diffusion of innovation which states that the probability of joining a group depends on the number of existing friends in the group and their connectedness.
3. To address our third statement we introduced Top Leaders – a method inspired by k-means – to mine communities in an information network. This method uses *iCloseness* to assign nodes to communities, which is effective in discovering communities and identifying outliers in a weighted or unweighted network. In Chapter 4 we applied the algorithm to known real world networks with ground truth as well as randomly generated networks and com-

pared the results against state-of-the-art community mining approaches. The experimental results confirm the accuracy and effectiveness of the proposed measure and our algorithm. *Top Leaders* requires k, the number of desired communities, as input. This may seem a major hurdle. However, it is possible to obtain k after running other algorithms such as FastModularity, SCAN or CFinder and provide the number of discovered communities to our algorithm. Given this parameter, *Top leaders* always outperforms the contenders in terms of quality of the communities as demonstrated in our experiments.

4. Finally we address our last statement in Chapter 5 and 6. There we illustrated the place and need for social network analysis in study of the interaction of users in e-learning environments. We then summarized some recent studies in this area. We then proposed Meerkat-ED, a specific and practical toolbox for analyzing students interactions in asynchronous discussion forums. Our toolbox prepares and visualizes overall snapshots of participants in the discussion forums, their interactions, and the leaders/peripheral students. Moreover, It creates a hierarchical summarization of the discussed topics, which gives the instructor a quick view of what is under discussion. It further illustrates individual student participation in these topics, measured by their centrality in the discussions on that topic, their number of posts, replies, and the portion of terms used by them. We believe exploiting the mining abilities of this toolbox would facilitate fair evaluation of students' participation in online courses.

7.2 Summary of Contributions

This MSc dissertation makes the following contributions:

1. A novel closeness measure, *iCloseness*, is presented in Chapter 3, which is inspired by the theory of Diffusion of Innovations. This measure is computed based on the intersection of neighbourhoods and quantifies the closeness between a node and a leader (i.e. most central node of the community).
2. A new, fast and accurate community mining approach is proposed in Chapter 3, named Top Leaders. Which applies *iCloseness* for mining communities in

a given weighted or unweighted network. Simply put, it regards a community as a set of followers congregating around a potential leader. Top Leaders starts by identifying promising leaders then iteratively assembles followers to their closest leaders to form communities, and subsequently finds new leaders in each group around which to gather followers again until convergence. Experimental results on real world and synthesized information networks verify the feasibility and effectiveness of our new community mining approach using *iCloseness*.

3. Meerkat-ED, a specific and practical toolbox for analyzing students interactions in online courses is proposed in Chapter 6. It applies social network analysis techniques including community mining to evaluate participation of students in asynchronous discussion forums, while its practical applicability is illustrated using our own case study data.

7.3 Future Research

Social Network analysis and its applications in different domains have attracted much attentions in recent years from researchers in data mining field. In this dissertation we have presented an algorithm to address one of the questions in social network analysis; community mining. We further illustrated one of its possible applicability in Education domain and for assessing the participation of students in online courses. There is much that could be done to extend the proposed work. For the first two contributions, future work could include:

1. Amending the Top Leaders algorithm:
   (a) The major drawback of the Top Leaders is that similar to k-means or k-medoid algorithms, it needs number of communities that should be detected as an input. This could be addressed by using the prior knowledge of the given network or some visualization tools to discover the number of communities. Moreover, this argument could be obtained using other community mining algorithms, while we have shown that Top Leaders improves the overall result. However, it is better to alter the

algorithm to find this parameter by itself, similar to different methods suggested for finding k in k-means e.g. choosing k using the silhouette [47] or information theoretic approaches [52].

(b) One interesting direction in modifying the Top Leaders algorithm is to make it to handle both attributes of the data entities and relation between them. The current version works only based on the relations between data entities, however, there could be data sets in which data entities not only are related but also have some specific attributes.

(c) Another generalization that could be considered for the Top Leaders algorithm is its adaptations for the case of directed networks.

(d) Top Leaders models a community as a leader and its followers, however, a community could be considered having more that one leader. This fuzzy notion of leadership could also be exploited in future works.

2. Regarding the educational application:

(a) For Meerkat-ED we can use a more thorough set of indicators prepared by a comprehensive requirement analysis of participation from educational point of view.

(b) In the current version of Meerkat-ED, each term community (topic) is represented using a set of top phrases of that community. A better way of representing the community, which could go as far as summarizing the content, is a part of our future work.

(c) Figuring out a systematic strategy for evaluating the Meerkat-ED, could also be another possible research direction. Currently it is not systematically evaluated though its practicability is illustrated by anecdotal evidence on one on-line course. It should be evaluated by requesting instructors of a certain number of on-line courses to evaluate the participation of their students with and without the use of MeerkatED and reporting its usefulness, merit and appeal using an elaborate questionnaire.

Bibliography

[1] Lada A. Adamic and Natalie Glance. The political blogosphere and the 2004 us election. In *WWW-2005 Workshop on the Weblogging Ecosystem*, 2005.

[2] Sitaram Asur, Srinivasan Parthasarathy, and Duygu Ucar. En event-based framework for characterizing the evolutionary behavior of interaction graphs. In *ACM SIGKDD)*, pages 913–921, 2007.

[3] Lars Backstrom, Dan Huttenlocher, Jon Kleinberg, and Xiangyang Lan. Group formation in large social networks: membership, growth, and evolution. In *ACM SIGKDD*, pages 44–54, 2006.

[4] S. Boccaletti, M. Ivanchenko, V. Latora, A. Pluchino, and A. Rapisarda. Detection of complex networks modularity by dynamical clustering. *Physical Review E*, 75:045102, 2007.

[5] Bela Bollobas. *Modern Graph Theory*. Springer, corrected edition, July 1998.

[6] Antonio Calvani, Antonio Fini, Marcello Molino, and Maria Ranieri. Visualizing and monitoring effective interactions in online collaborative groups. *British Journal of Educational Technology*, 2009.

[7] P.K. Chan, M. D. F. Schlag, and J. Y. Zien. Spectral k-way ratio-cut partitioning and clustering. In *Proceedings of the 30th International Conference on Design Automation*, pages 749–754, 1993.

[8] J. Chen, O. R. Zaïane, and R. Goebel. Detecting communities in social networks using max-min modularity. In *Proceedings of the SIAM Data Mining Conference*, pages 105–112, 2009.

[9] J. Chen, O. R. Zaïane, and R. Goebel. Local community identification in social networks. In *Proceedings of the International Conference on Advances in Social Network Analysis and Mining(ASONAM)*, pages 237–242, 2009.

[10] J. Chen, O. R. Zaïane, and R. Goebel. A visual data mining approach to find overlapping communities in networks. In *Proceedings of the International Conference on Advances in Social Network Analysis and Mining(ASONAM)*, pages 338–343, 2009.

[11] A. Clauset. Finding local community structure in networks. *Physical Review E*, 72:026132, 2005.

[12] A. Clauset, M. E. J. Newman, and C. Moore. Finding community structure in very large networks. *Phys. Rev. E*, 70:066111, 2004.

[13] Thanasis Daradoumis, Alejandra Martínez-Monés, and Fatos Xhafa. A layered framework for evaluating on-line collaborative learning interactions. *Int. J. Hum.-Comput. Stud.*, 64(7):622–635, 2006.

[14] R H Davis. Social network analysis - an aid in conspiracy investigations. *FBI Law Enforcement Bulletin*, 50(12):11–19, December 1981. The use of social network analysis in the conduct of investigations of conspiracies is described.

[15] Maarten de Laat, Vic Lally, Lasse Lipponen, and Robert-Jan Simons. Investigating patterns of interaction in networked learning and computer-supported collaborative learning: A role for social network analysis. *International Journal of Computer-Supported Collaborative Learning*, 2(1):87–103, March 2007.

[16] I. S. Dhillon, S. Mallela, and D. S. Modha. Information-theoretic coclustering. In *KDD*, pages 89–98, 2003.

[17] Erlin, Norazah Yusof, and Azizah A. Rahman. Students' interactions in online asynchronous discussion forum: A social network analysis. In *International Conference on Education Technology and Computer*, pages 25–29, Los Alamitos, CA, USA, April 2009. IEEE Computer Society.

[18] Martin Ester, Hans-Peter Kriegel, Jorg Sander, and Xiaowei Xu. A density-based algorithm for discovering clusters in large spatial databases with noise. In *ACM SIGKDD)*, pages 226–231, 1996.

[19] G. W. Flake, S. Lawrence, and C. L. Giles. Efficient identification of web communities. In *KDD*, pages 150–160, 2000.

[20] Santo Fortunato. Community detection in graphs. *Physics Reports*, 486(3-5):75 – 174, 2010.

[21] Lise Getoor and Christopher P. Diehl. Link mining: a survey. *ACM SIGKDD Explorations*, 7(2):3–12, 2005.

[22] Michelle Girvan and M. E. J. Newman. Community structure in social and biological networks. In *Proceedings of the National Academy of Science USA, 99:8271-8276*, 2002.

[23] Benjamin H. Good, Yves A. de Montjoye, and Aaron Clauset. Performance of modularity maximization in practical contexts. *Physical Review E*, 81(4):046106+, Apr 2010.

[24] S. Gregory. An algorithm to find overlapping community structure in networks. In *PKDD*, pages 91–102, 2007.

[25] A. Gruzd and C. Haythornthwaite. Automated discovery and analysis of social networks from threaded discussions. In *International Network of Social Network Analysis (INSNA) Conference*, St. Pete Beach, Florida, January 2008.

[26] Anatoliy Gruzd and Caroline A. Haythornthwaite. The analysis of online communities using interactive content-based social networks. extended abstract. In *Proceedings of the American Society for Information Science and Technology (ASIS&T) Conference*, pages 523–527, Columbus, OH, USA, October 2008.

[27] Anatoliy A. Gruzd. *Automated Discovery of Social Networks in Online Learning Communities*. PhD thesis, University of Illinois at Urbana-Champaign, June 2009.

[28] S. Hrastinski. What is online learner participation? a literature review. *Computers & Education*, 51(4):1755–1765, December 2008.

[29] D. Jensen. Statistical challenges to inductive inference in linked data, 1999.

[30] G. Karypis and V. Kumar. Multilevel k-way partitioning scheme for irregular graphs. *Journal of Parallel and Distriuted Computing*, 48(1):96–129, 1998.

[31] Matt J. Keeling and Ken T. Eames. Networks and epidemic models. *Journal of the Royal Society, Interface / the Royal Society*, 2(4):295–307, September 2005.

[32] B. W. Kernighan and S. Lin. An efficient heuristic procedure for partitioning graphs. *Bell System Technical Journal*, 49:291–307, 1970.

[33] Valdis Krebs. Books about us politics. http://www.orgnet.com/.

[34] Laghos and Zaphiris. Sociology of student-centred e-learning communities: A network analysis. In *IADIS international conference*, Dublin, Ireland, July 2006. e-Society.

[35] Andrea Lancichinetti, Santo Fortunato, and Filippo Radicchi. Benchmark graphs for testing community detection algorithms. *Physical Review E*, 78:046110, 2008.

[36] F. Luo, J. Z. Wang, and E. Promislow. Exploring local community structures in large networks. In *WI '06: Proceedings of the 2006 IEEE/WIC/ACM International Conference on Web Intelligence*, pages 233–239, 2006.

[37] Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze. *Introduction to Information Retrieval*. Cambridge University Press, 1 edition, July 2008.

[38] Wellman B. Marin, A. *Handbook of Social Network Analysis*, chapter Social Network Analysis: An Introduction. Sage, forthcoming, 2010.

[39] T. Nepusz, A. Petroczi, L. Negyessy, and F. Bazso. Fuzzy communities and the concept of bridgeness in complex networks. *Physical Review E*, 77, 2008.

[40] M. E. J. Newman. Detecting community structure in networks. *Eur. Phys. J.B*, 38:321–330, 2004.

[41] M. E. J. Newman. Modularity and community structure in networks. *PROC.NATL.ACAD.SCI.USA*, 103, 2006.

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

[43] A. Ng, M. Jordan, and Y. Weiss. On spectral clustering: Analysis and an algorithm. In *NIPS*, pages 849–856, 2001.

[44] Kari Nurmela, Erno Lehtinen, and Tuire Palonen. Evaluating cscl log files by social network analysis. *Computer Support for Collaborative Learning*, 1999.

[45] Pajek. http://vlado.fmf.uni-lj.si/pub/networks/pajek/.

[46] G. Palla, I. Derenyi, I. Farkas, and T. Vicsek. Uncovering the overlapping community structure of complex networks in nature and society. *Nature*, 435:814–818, 2005.

[47] Peter Rousseeuw. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *J. Comput. Appl. Math.*, 20(1):53–65, 1987.

[48] Jorge M. Santos and Mark Embrechts. On the use of the adjusted rand index as a metric for evaluating supervised classification. In *ICANN (2)*, pages 175–184, 2009.

[49] Purnamrita Sarkar and Andrew Moore. Dynamic social network analysis using latent space models. *ACM SIGKDD Explorations: Special Edition on Link Mining*, 7(2):31–40, 2005.

[50] J. Shi and J. Malik. Normalized cuts and image segmentation. *IEEE. Trans. on Pattern Analysis and Machine Intelligence*, 2000.

[51] David Strang and Soule Sarah A. Diffusion in organizations and social movements: From hybrid corn to poison pills. *Annual Review of Sociology*, 24(1):265–290, 1998.

[52] Catherine A. Sugar and Gareth M. James. Finding the number of clusters in a dataset: An information-theoretic approach. *Journal of the American Statistical Association*, 98:750–763, 2003.

[53] C. Tantipathananandh, T. Berger-Wolf, and D. Kempe. A framework for community identification in dynamic social networks. In *ACM SIGKDD*, pages 717–726, 2007.

[54] S. Wasserman and K. Faust. Social network analysis: Methods and applications, Cambridge University Press, 1994.

[55] P. A. Willging. Using social network analysis techniques to examine online interactions. *US-China Education Review*, 2(9):46–56, 2005.

[56] X. Xu, N. Yuruk, Z. Feng, and T. A. J. Schweiger. Scan: a structural clustering algorithm for networks. In *ACM SIGKDD*, pages 824–833, 2007.

[57] W. W. Zachary. An information flow model for conflict and fission in small groups. *Journal of Anthropological Research*, 33:452–473, 1977.

[58] S. Zhang, R.-S. Wang, and X.-S. Zhang. Identification of overlapping community structure in complex networks using fuzzy c-means clustering. *Physica A*, 374:483–490, 2007.