Available online at www.sciencedirect.com

SciVerse ScienceDirect

Procedia - Social and Behavioral Sciences 22 (2011) 49 – 58

Procedia
Social and Behavioral Sciences

Community Evolution Mining in Dynamic Social Networks

Mansoureh Takaffoli, Farzad Sangi, Justin Fagnan, Osmar R. Zaïane*

Department of Computing Science, University of Alberta,
Edmonton, Alberta, Canada T6G 2E8

Abstract

Data that encompasses relationships is represented by a graph of interconnected nodes. Social network analysis is the study of such graphs which examines questions related to structures and patterns that can lead to the understanding of the data and predicting the trends of social networks. Static analysis, where the time of interaction is not considered (i.e., the network is frozen in time), misses the opportunity to capture the evolutionary patterns in dynamic networks. Specifically, detecting the community evolutions, the community structures that changes in time, provides insight into the underlying behaviour of the network. Recently, a number of researchers have started focusing on identifying critical events that characterize the evolution of communities in dynamic scenarios. In this paper, we present a framework for modeling and detecting community evolution in social networks, where a series of significant events is defined for each community. A community matching algorithm is also proposed to efficiently identify and track similar communities over time. We also define the concept of meta community which is a series of similar communities captured in different timeframes and detected by our matching algorithm. We illustrate the capabilities and potential of our framework by applying it to two real datasets. Furthermore, the events detected by the framework is supplemented by extraction and investigation of the topics discovered for each community.

Keywords: Dynamic Social Networks ; Community Evolution ; Dynamic Network Analysis ; Evolutionary Analysis

1. Introduction

We define social networks as information networks that are represented by graphs and depict the interactions between individuals or entities. In these networks, each individual is represented by a node in the network, and there is an edge between two nodes if an interaction has occurred, or a relationship exists, between the two individuals during the observation time. For instance, the exchange of ideas, information, and experiences between people in the web can be modeled as a social network.

The analysis of social networks is of interest to many fields such as sociology (Wasserman & Faust, 1994), epidemiology (Meyers et al., 2006), recommendation systems (Palau et al., 2004), email communication (Tyler et al., 2003), criminology (Calvo-Armengol & Zenou, 2003), etc. The need to identify communities, which are densely connected subset of individuals that are loosely connected to others (Newman & Girvan, 2004), has recently driven significant attention in the research community. The analysis of communities can help determine the structural properties of the networks as well as facilitate applications such as targeted marketing and advertising (Kempe et al., 2003), and finding influential individuals (Berger-Wolf & Saia, 2005).

1877–0428 © 2011 Published by Elsevier Ltd. Open access under CC BY-NC-ND license.
doi:10.1016/j.sbspro.2011.07.055

Most networks, such as social media, blogs, and co-authorship networks, are dynamic as they tend to evolve gradually, due to frequent changes in the activity and interaction of their individuals (Newman & Park, 2003). Furthermore, the communities inside a dynamic network could grow or shrink, and the community membership of the individuals shifts regularly (Backstrom et al., 2006; Leskovec et al., 2005). In these dynamic networks, researchers may be interested in the evolution of communities and membership of individuals such as author communities in the blogosphere (Lin et al., 2008), the analysis of mobile subscriber networks (Wu et al., 2009), and evolution of research communities (Palla et al., 2007). However, past community detection analyses of social networks modeled the dynamic network as a static graph by discarding the temporal information. This static representation misses the opportunity to detect the evolutionary behavior of the network and the communities.

One way to model the structural changes in dynamic networks is to convert an evolving network into static graphs at different snapshots (Berger-Wolf & Saia, 2006). Such dynamic analysis of social network, especially assessing the evolution of communities, provides various insights into: 1) understanding the structures of the complex networks; 2) detecting a drastic change in the interaction patterns; 3) making predictions on the future trends of the network, etc. The evolution of communities in dynamic social networks can be tracked by identifying critical events that characterize the changes in a community over time (Palla et al., 2007; Asur et al., 2007; Greene et al., 2010; Takaffoli et al., 2010, 2011). In this paper, we propose a two-stage framework in order to analyze the dynamic evolution of communities. In this framework, we assume that in each snapshot the communities are independently extracted, thus, our framework is independent of the community mining algorithms. In the first stage of the framework, a one-to-one matching algorithm, based on weighted bipartite matching (Kuhn, 1955), is proposed to match the communities extracted at different time steps. Then, a meta community is constructed for each series of similar communities detected by the matching algorithm in different timeframes. The Meta community provides the evolution of its constituent communities. In the second stage, a collection of significant events are identified and used to explain the differences between the communities of a meta community over time.

In order to evaluate our proposed framework, we consider two social network datasets: The Enron dataset, which provides emails between employees of the Enron Corporation; and the DBLP dataset, which contains a computer science co-authorship network. Our results are supplemented by an automatic extraction of the topics for each community to provide more reasonable results in terms of events discovered vis-*à*-vis the topics of discourse. The rest of the paper is organized as follows: In the next section, we provide a brief overview of existing research in the area of dynamic community mining. The problem formulation is described in Section 3, followed by the explanation of events in Section 4. Section 5, describes a community matching algorithm. The evaluation of the proposed algorithm on two real datasets is given in Section 6. Section 7 concludes with a summary and suggestions for future work.

2. Related Work

There has been a considerable amount of work done to detect communities in static graphs, such as modularity methods (Newman, 2006; Newman & Girvan, 2004; Chen et al., 2009), spectral clustering methods (White & Smyth, 2005), stochastic methods (Handcock et al., 2007; Airoldi et al., 2008; Choi et al., 2010), and heterogeneous clustering methods (Sun et al., 2009a,b; Doreian et al., 2004, 2005). Although most social networks evolve gradually (Newman & Park, 2003), these methods model the dynamic network as a static graph by removing information about the time of the interactions.

Recently, the temporal evolution of social networks has attracted many researchers. White et al. (1976) is the first proposed approach for finding community structure applied to network observed over time as well as over different relations. Leskovec et al. (2005) study the patterns of growth for large social networks based on the properties of large networks, such as the degree of distribution and the small-world phenomena. They also propose a graph generation model to produce networks satisfying the discovered patterns. Backstrom et al. (2006) approximate the probability of an individual joining two explicitly defined communities based on defining critical factors and then analyze the evolution of these communities. Kumar et al. (2006) provide the properties of two real-world networks and then analyze the evolution of structure in these networks. However, in these cases the properties on the graph level are studied while the properties on the level of communities are not observed.

Berger-Wolf & Saia (2006) propose a mathematical and computational framework that enables tracking the evolution of communities. The same team in their later work (Tantipathananandh et al., 2007) formulate the detection of dynamic communities as a graph colouring problem. They provide a heuristic technique that involves greedily

matching detected communities at different snapshots. However, they only consider matching across two consecutive snapshots and focus mostly on tracking the membership of a given individual. Falkowski et al. (2006) discover the evolution of communities by applying clustering on a graph formed by all detected communities at different time points.

A number of researchers are working on identifying critical events that characterize the evolution of communities in dynamic social networks. Palla et al. (2007) identify events by applying Clique Percolation Method (CPM) community mining (Palla et al., 2005) on a graph formed by the communities discovered at two consecutive snapshots. Then, based on the results of the community mining algorithm, events pertaining to the communities are specified. Asur et al. (2007) define critical events between detected communities at two consecutive snapshots which are implement in the form of bit operations. However, these events do not cover all of the transitions that may occur for a particular community. Takaffoli et al. (2010) provide an event-based framework to capture all of the transitions between communities at two consecutive snapshots. In a later work (Takaffoli et al., 2011), the event definition formula is improved to track the transitions of communities over the entire observation time, not only between two consecutive snapshots. Moreover, the results are validated via the extraction of the topics for each community. Greene et al. (2010) describe a weighted bipartite matching to map communities and then characterized each community by a series of events.

All of the above work focus on analyzing the evolution of communities by using a two-stage approach. In this approach the communities are first detected independently for each snapshot, and then compared to determine the evolution. This two-stage approach is suitable for the social networks with highly dynamic community structures. Another approach is to use evolutionary community mining, where the community mining at a particular time is influenced by the communities detected in previous time. Thus, this time-dependent community mining approach finds a sequence of communities with temporal similarity and hence, is suitable for networks with community structures that are more stable over time. Sarkar & Moore (2005) develop the Latent space model with temporal change to find communities that are consistent with the network at the current time and with the communities detected at a previous time. Mucha et al. (2010) generalize the Laplazian dynamics approach in order to extend modularity maximization to study community structure across multiple times in dynamic social network. Time-dependent community mining approach, which considers both current and historic information into the objective of the mining process, is also proposed (Chakrabarti et al., 2006; Lin et al., 2008; Falkowski et al., 2008; Tang et al., 2008; Asur & Parthasarathy, 2009; Sun et al., 2010). However, this approach cannot detect the evolution of communities and events related to them in dynamic social network with explicitly defined communities. Furthermore, most of the algorithms using time-dependent community mining approach can only discover small changes in communities in consecutive time frames and any drastic change in short time remains undetected.

Analogous to community mining and related to our work there are recent studies on the evolution of clusters. Ganti et al. (2002) propose a change detection framework called FOCUS. In FOCUS, two datasets are compared by computing a deviation measure between them. Chakrabarti et al. (2006) discover clusters and their transitions at the same time. They obtain high-quality clusters at the current timestamp while maintaining similarity with the clusters identified in previous timestamps. Spiliopoulou et al. (2006) propose an event-based framework called MONIC to model and track cluster transitions. They also introduce the concept of cluster matching to simplify the detection and evaluation of the cluster events that occurred. Oliveira & Gama (2010) tackle the problem of monitoring the transitions experienced by clusters over time by identifying the temporal relationships among them.

In summary, none of the previous work cover all of the changes a community may experience during the observation time of a dynamic social network. However, finding patterns of interaction and predicting the future structure of communities is attractive for many areas such as disease modelling (Eubank et al., 2004), information transmission (Kempe et al., 2003; Tyler et al., 2003), and business management (Bernstein et al., 2002), but is only possible by capturing all the transitions of the communities in the dynamic social network. Thus, we propose a two-stage approach to analyze the dynamic evolution of communities and detect all the events related to the communities. These events can be used as a building block to predict the future structure of communities by discovering the patterns of events in the dynamics of the network.

3. Problem Formulation

We model the dynamic social network as a sequence of graphs $\{G_1, G_2, ..., G_n\}$, where $G_i = (V_i, E_i)$ represents a graph with only the set of individuals and interactions at a particular snapshot i. Unlike previous approaches (Palla

(a) (b) (c)

Figure 1: Examples to illustrate the similarity measure: (a) Two communities with 110 and 120 members where they have 30 mutual members; (b) Two communities with 100 and 30 members where they have 20 mutual members; (c) Two communities with 100 and 40 members where they have 40 mutual members.

et al., 2007; Tantipathananandh et al., 2007; Falkowski et al., 2006), the communities at any snapshot can be the result of any static community mining algorithm. The n_i communities detected at the ith snapshot are then denoted by $C_i = \left\{C_i^1, C_i^2, ..., C_i^{n_i}\right\}$, where community $C_i^p \in C_i$ is also a graph denoted by (V_i^p, E_i^p). In this paper, we distinguish between the terms community and meta community. A community contains individuals that are densely connected to each other at a particular time snapshot. On the other hand, a meta community is a series of similar communities at different time snapshots and represents the evolution of its constituent communities ordered by time of the snapshots. Here, the meta community is denoted by $M = \{c_{t_1}, c_{t_2}, ..., c_{t_m}\}$, where $t_1 < t_2 < ... < t_m$, and c_{t_i} represents its instance community at time t_i. Meta community M with lifetime m contains m communities and any of its two consecutive communities are similar with respect to specific k. In this paper, we reduce the problem of detecting the transition and evolution of communities to identify meta communities and also the events characterizing the changes of the communities across the time of observation.

In the literature there are different taxonomies to categorize the changes of clusters, communities, or patterns that evolve over time (Spiliopoulou et al., 2006; Oliveira & Gama, 2010; Asur et al., 2007; Palla et al., 2007). In this work, to capture the changes that are likely to occur for a community, we consider five events including *split*, *survive*, *dissolve*, *merge*, and *form*. A community may *split* at a later snapshot if it fractures into multiple communities. It can *survive* if there exists a similar community in a future snapshot. In the case where there is no similar community at a later snapshot, then the community *dissolves*. A set of communities may also *merge* together at a later snapshot. Furthermore, at any snapshot there may be newly *formed* communities, where there is no similar community at a previous snapshot. The meta community can then be interpreted as a sequence of communities ordered by time, from the timeframe where it first appears to the timeframe where it is last observed.

The key concept for the detection of the events, and also the meta community, is the concept of similarity between communities at different times. Two communities that are discovered at different snapshots are similar if a certain percentage, $k \in [0, 1]$, of their members are mutual. The similarity threshold k captures the tolerance to member fluctuation, and can be set based on the characteristic of the underlying network. A high similarity threshold would be expected in a network with stable communities that have many members who participate over a long time and less fluctuating members. In highly dynamic social networks, where the structure changes over time, there are unstable communities such that the members of a community leave gradually while new ones join. This community may exist for a long time, even if all of its original individuals have left. Thus, to identify groups that make up this unstable community, a low similarity threshold would be preferred. The formal definition of similarity between two communities is defines as follows:

Community Similarity: Let C_i^p and C_j^q be the community detected at snapshot i and j respectively ($i \neq j$). The two communities C_j^q and C_i^p are similar if and only if their shared members make up at least k proportion of the biggest community:

$$\text{sim}(C_i^p, C_j^q) = \begin{cases} \frac{|V_i^p \cap V_j^q|}{\max(|V_i^p|, |V_j^q|)} & if \quad \frac{|V_i^p \cap V_j^q|}{\max(|V_i^p|, |V_j^q|)} \geq k \\ 0 & otherwise \end{cases} \tag{1}$$

Dividing the number of members that exist in both communities by the size of the biggest community (Equation 1) scales for different sizes of communities. Figure 1a illustrates an example when two communities are about the same size. These two communities shared 30 members, thus, they are similar if the similarity threshold k is less than 0.2

(i.e., $\frac{30}{120} \geq 0.25$). Figure 1b shows an example when one of the communities is considerably smaller than the other and they have 20 mutual members. Hence, setting $k \leq 0.2$ marks them as two similar communities since $\frac{20}{100} \geq 0.2$. An example when one community contains all the members of the other community is shown in Figure 1c. The mutual members of these two communities are 40 individuals, thus, with $k \leq 0.4$ these communities are similar (i.e., $\frac{40}{100} \geq 0.4$). The choice of the similarity threshold k is depend on the characteristic of the underlying network. In the following sections, we provide an algorithm to determine k for an arbitrary dynamic social network.

In order to detect the meta communities and the events, the set of communities at a given snapshot have to be matched to the communities at previous snapshots based on their similarity. However, since a community may have similarity with several communities at the same time, the matching process becomes non-trivial. The optimization problem that arises here is to find a match that maximizes the pair wise similarity over all selected matches, not only the direct preceding snapshot but potentially other previous snapshots.

4. Dynamic Community Analysis

There are several taxonomies that define the fundamental events that can be used to characterize the evolution of dynamic communities (e.g. Spiliopoulou et al., 2006; Asur et al., 2007; Takaffoli et al., 2010). Given the definition of meta communities, similarity, and similarity threshold k, a community C_i^p at the ith snapshot may undergo different transitions in later snapshots. Community C_i^p *splits* at snapshot $j > i$ if it fractures into multiple communities with at least k proportion of their members from C_i^p. Community C_i^p *survives* if there is a community C_j^q at snapshot $j > i$ such that their meta communities are identical. In the case where there is no such community, C_i^p *dissolves*. Only the *survive* and *dissolve* events are mutually exclusive, while the *split* event can be combined with the other two. Community C_i^p *splits* and *survives* at the jth snapshot if it fractures into more than one community and one of these communities has the same meta community as C_i^p. Community C_i^p *splits* and *dissolves* at the jth snapshot if it fractures into other communities and none of these communities have the same meta community as C_i^p.

In addition to the three events mentioned above, a set of communities in C_i can *merge* together in community C_j^q at snapshot $j > i$. The *merge* event occurs when at least k proportion of the members from multiple communities in C_i, exist in C_j^q. Furthermore, at any snapshot there may be newly *formed* communities. These communities are the ones that do not belong to any of the already existing meta communities.

In the following, the formal definitions of these events are provided, where match(C_i^p, j) denotes the optimal match for C_i^p at jth snapshot which is the results of the optimal matching algorithm for community C_i^p:

Form: A community C_i^p *forms* at ith snapshot if there is no community match for it in any of the previous snapshots:

$$\text{form}(C_i^p, i) = 1 \quad iff \quad \forall j < i \quad \text{match}(C_i^p, j) = \emptyset \tag{2}$$

Dissolve: A community C_i^p *dissolves* at ith snapshot if there is no community match for it in any of the next snapshots:

$$\text{dissolve}(C_i^p, i) = 1 \quad iff \quad \forall j > i \quad \text{match}(C_i^p, j) = \emptyset \tag{3}$$

Survive: A community C_i^p *survives* at ith snapshot if there exists a snapshot $j > i$ that contains a community match for C_i^p:

$$\text{survive}(C_i^p, i) = 1 \quad iff \quad \exists j > i \quad and \quad \exists C_j^q \in C_j \quad \text{match}(C_i^p, j) = C_j^q \tag{4}$$

Split: A community C_i^p at ith snapshot *splits* to a set of communities $C_j^* = \left\{C_j^1, ..., C_j^n\right\}$ at snapshot $j > i$ if at least k proportion of the members of the communities in C_j^* are from community C_i^p. Also in order to prevent the case where most of the members of C_i^p leave the network, the mutual members of the union of the communities in C_j^* with C_i^p should be greater than k proportion of C_i^p:

$$\begin{aligned} &\text{split}(C_i^p, i) = 1 \quad iff \quad \exists j > i \quad and \quad \exists C_j^* = \left\{C_j^1, ..., C_j^n\right\} \in C_j \quad where \\ &1) \forall C_j^r \in C_j^* \quad \frac{|V_i^p \cap V_j^r|}{|V_j^r|} \geq k \qquad 2) \frac{|(V_j^1 \cup V_j^2 ... \cup V_j^n) \cap V_i^p|}{|V_i^p|} \geq k \end{aligned} \tag{5}$$

Merge: A set of communities $C_i^* = \left\{C_i^1, ..., C_i^n\right\}$ at ith snapshot *merge* to C_j^q at snapshot $j > i$ if C_j^q contains at least k proportion of the members from each community in C_i^*. Also to prevent the case where most of the members of C_j^q did not exist before, the mutual members of the union of all communities in C_i^* with C_j^q should be greater than k proportion of C_j^q:

$$\begin{aligned} &\text{merge}(C_i^* = \left\{C_i^1, ..., C_i^n\right\}, i) = 1 \quad iff \quad \exists j > i \quad and \quad \exists C_j^q \quad where \\ &1)\forall C_i^r \in C_i^* \quad \frac{|V_i^r \cap V_j^q|}{|V_i^r|} \geq k \qquad 2)\frac{|(V_i^1 \cup V_i^2 ... \cup V_i^n) \cap V_j^q|}{|V_j^q|} \geq k \end{aligned} \tag{6}$$

5. Community Matching Algorithm

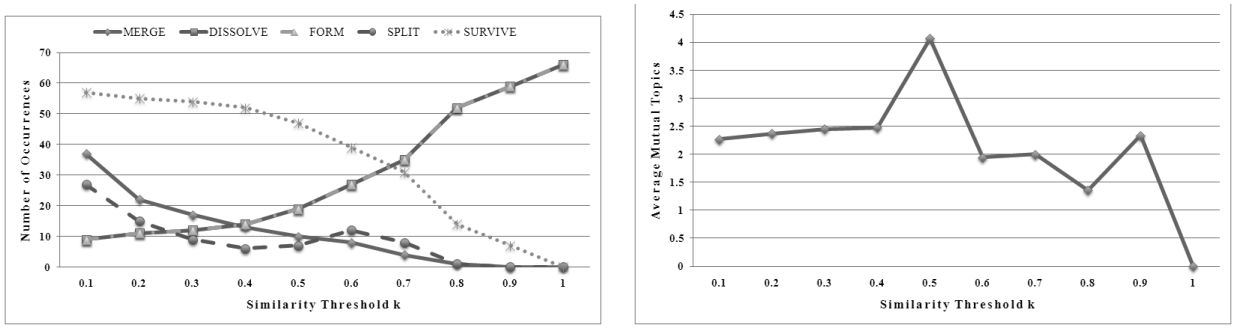
After the communities have been extracted at each time step, the set of communities at consecutive snapshots have to be matched with each other. This raises the problem of finding the match of a given community at time t among the communities at time $t-1$. A simple approach would be to match communities from consecutive time steps in descending order of their similarity. Since a community may overlap with several other communities at the same time, the matching of the communities based on solely their similarity becomes a complicated process. Furthermore, a community may not necessarily be observed at all the snapshots after its *formation* and may be missing from one or more snapshots. This reflects that, although a community was absent, after a few snapshots it may suddenly reappear in the network. To consider this scenario, a matching between communities at time t and all of the other communities at time $t' < t$ could be considered. However, this solution is computationally expensive and it may lead to noisy results in scenarios where individuals participate in several communities but are active in different communities at different snapshots.

In this paper, we propose a matching algorithm that maximizes the amount of similarity preserved from one snapshot to the next, while considering the case of absent communities.We obtain series of weighted bipartite graphs by proceeding in increasing order of time steps. Initially, each community at snapshot 0 is considered as a newly *formed* community and a new meta community is created for each of them. In iteration t, in order to give more chance to similar communities in close temporal proximity, we first construct a weighted bipartite graph between communities at snapshot t and communities at $t-1$. The weight between communities is calculated with the notion of similarity introduced before. Then, the maximum weight bipartite matching (Kuhn, 1955) is applied to connect communities at time t to communities at time $t-1$. If community C_t^p matches to community C_{t-1}^q, then we can say that C_t^p is the *survival* of C_{t-1}^q (i.e. C_{t-1}^q *survives* to C_t^p). Thus, the C_t^p is added to the meta community that constitutes community C_{t-1}^q. For the communities at time t which are left with no counterpart from C_{t-1}, another bipartite matching is constructed between them and the communities at time $t-2$ whose meta communities have not been selected yet. Once again, the maximum weight bipartite matching is applied to detect *survival* events and also to update meta communities. The process of constructing the bipartite graph is continued until all the communities at time t match a community at time $0 \leq t' < t$ or all existing meta communities are already taken. We consider the communities left with no matches from $\{C_0, ..., C_{t-1}\}$ as newly *formed* communities and build a new meta community for each of them. After every community at time t is assigned to one meta community, we move to the next iteration.

The meta communities detected by the above algorithm represent the evolution of its constituent communities ordered by time of the snapshots. The last community of every meta community is marked as *dissolve* since it is unmatched for all of the next snapshots.

6. Experiments

In this section, we validate the effectiveness and feasibility of our algorithms through experiments on two real datasets. The first test bench is the Enron email dataset which is the message exchange of people within the Enron Corporation, while the second dataset is a subset of DBLP, which is a computer science bibliography dataset. On both these datasets, we gain insights on the impact of the similarity threshold on the evolution of the communities. Furthermore, for each dataset the optimal similarity threshold is selected by automatic extraction and the investigation of the topics of communities. Our framework is integrated into Meerkat (Chen et al., 2010), which enables us to preview the graph of each timeframe and the events for a particular community. As noted previously, our dynamic evolution detection is independent of the choice of the underlying static community mining algorithm. Due to computational



(a) (b)

Figure 2: Experiments on Enron email dataset: (a) Communities transitions for different values of similarity threshold k; (b) The average topic continuation for different values of similarity threshold k.

efficiency, we apply the local community mining algorithm (Chen et al., 2009), that is also implemented in Meerkat to produce sets of disjoint communities for each snapshot.

6.1. Enron email dataset

The well known Enron dataset is chosen to test whether our framework can detect interesting events in real networks. Particularly, we extract 250 email addresses from the original dataset, and consider the email exchanges between these addresses. Moreover, we only consider the last year of Enron Corporation (2001) to create monthly snapshots. For each of the 12 snapshots, one graph is constructed from email exchanges between these extracted addresses. In our experiments, we investigate the impact of similarity threshold k on the community evolution. The similarity threshold is varied from 0.1 to 1.0 in steps of 0.1 and the number of events occurring during the 12 snapshots is counted for each k step (Figure 2a). We see from Figure 2a that the choice of the similarity threshold k has a noticeable effect on the transitions of communities: the number of *survival*, *merge*, and *split* events drops as k increases, while there are more *dissolution* and *formation*. With low values of k, more communities are matched together, thus, we can observe a significant number of *surviving* communities. On the other hand, high values of k result in more a conservative matching behaviour and short-life communities.

The question that arises here is which similarity threshold results in the most appropriate community evolutions for the Enron dataset. In order to evaluate the community evolutions and select the optimal similarity threshold, we incorporate automatically extracted topics for each community. The Keyphrase Extraction Algorithm (KEA) (Witten et al., 1999) is applied to produce a list of the keywords discussed in the emails within each community. The topics for each community corresponds to its 10 most frequent keywords, as extracted by KEA. We expect that a community which *survives* multiple timeframes is more likely to continue discussions of the same topics (Takaffoli et al., 2011). Topics that persist in a community from one snapshot to the other are called mutual topics. The average mutual topics between any two *survival* communities during the observation time for different similarity thresholds is shown in Figure 2b. The survival communities mostly discuss the same topics, thus, the similarity threshold that corresponds to the highest average mutual topics illustrates the transition of the communities better than the others. Figure 2b shows that as k increases from 0.1 to 0.5 the average number of mutual topics also increases. However, for $k > 0.5$ the number of mutual topics decreases sharply. Based on this observation, we can conclude that for the Enron dataset the choice of $k = 0.5$ results in the most meaningful community transitions since it has highest average mutual topics. It is worth mentioning that for some applications, more than one k may results in maximum mutual topics. In this case based on the characteristic of the social network, one of the k that results in the maximum mutual topics is selected: A high similarity threshold k is required for the networks with rather stable communities, while a low similarity threshold k is selected for the network with highly dynamic communities. Furthermore, the appropriate similarity threshold might also be defined based on the theoretical considerations.

As mentioned before, in this paper a community refers to individuals that are densely connected to each other at a particular snapshot, while a meta community is a series of similar communities at different snapshots and represents the evolution of its constituent communities ordered by time of the snapshots. After determining the appropriate

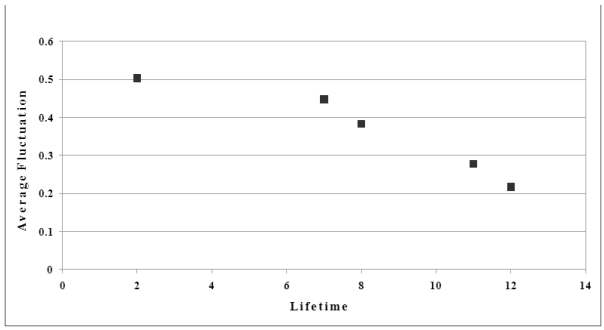
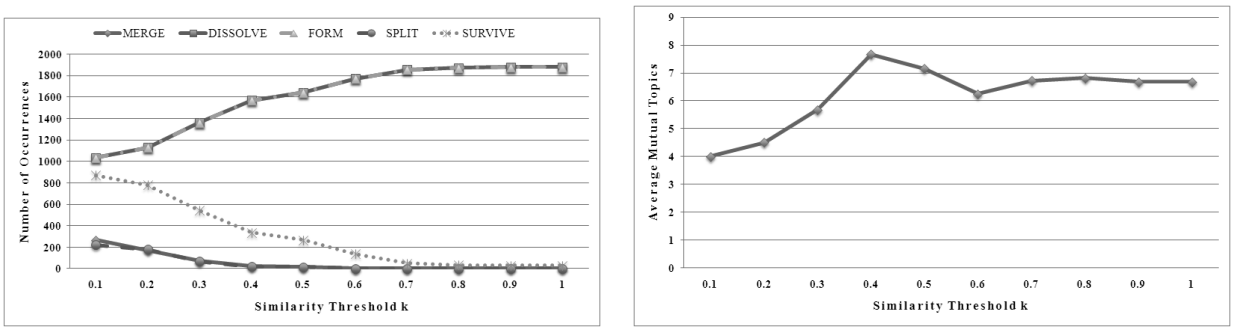


Figure 3: Average members fluctuation and lifetime of Enron meta community.



(a) (b)

Figure 4: Experiments on DBLP co-authorship dataset: (a) Communities transitions for different values of similarity threshold k; (b) The average topic continuation for different values of similarity threshold k.

similarity threshold for the Enron dataset, we consider two basic quantities characterizing a meta community: its member fluctuation and its lifetime. The lifetime of a meta community represents the number of snapshots between the *formation* and *dissolution* of it (i.e. the time difference between its first instantiation and its last instantiation), while member fluctuation characterizes the average similarity of its constituent communities between consecutive snapshots. Consider a meta community $M = \{c_{t_1}, c_{t_2}, ..., c_{t_m}\}$ with lifetime m containing m communities, where $t_1 < t_2 < ... < t_m$, and c_{t_i} represents its instance community at time t_i. We define the member fluctuation of the meta community M as:

$$\text{fluctuation}(M) = 1 - \frac{\sum_{i=1}^{m-1} \text{sim}(c_{t_i}, c_{t_{i+1}})}{m-1} \tag{7}$$

We observe an interesting effect when investigating the relationship between the lifetime and the members fluctuation. In Figure 3 the average lifetime as a function of the members fluctuation is depicted for similarity threshold $k = 0.5$. The figure indicates that meta communities with average low fluctuating members usually tend to live longer, while meta communities with higher fluctuation *dissolve* sooner.

6.2. DBLP co-authorship dataset

We extract our next dataset from DBLP where the co-authorship network for three major data mining conferences: ICDM, SIGMOD, and KDD from 2004 to 2009 is chosen. The resulting network contains 7500 individuals where the duration of a snapshot is defined to be one year. The impact of the similarity threshold k on the community evolution is also investigated for the DBLP network (Figure 4a). Again, we observe that the choice of the similarity threshold k has a noticeable effect on the transitions of communities. Low values of k lead to a significant number of *surviving* communities, while high values of k results in more *dissolutions* and short-life communities.

The evolution of communities in the DBLP dataset is also validated by extracting the topics of the papers published within communities. The average mutual topics between two *survival* communities for different similarity threshold is shown in Figure 4b. We can observe that setting $k = 0.4$ in the DBLP dataset results in the highest mutual topics between *survival* communities. Furthermore, for similarity threshold $k = 0.4$, again the same trend as in the Enron dataset can be observed where meta communities with average low fluctuating members have longer lifetime. The difference between the optimal similarity threshold of the DBLP and the Enron dataset is due to their different structure: the Enron email dataset has rather stable communities with a considerable amount of members who participate over a long time and a small amount of fluctuating members. Thus, a high similarity threshold ($k = 0.5$) is required. On the other hand, in the DBLP co-authorship network, communities can be highly dynamic where members leave gradually, while new ones join. Hence, a rather low similarity threshold ($k = 0.4$) is used to analyze the evolution of communities in this network.

7. Conclusion

In this paper, we present a framework for the monitoring of community transitions and evolutions over time. Our framework encompasses community matching algorithm and also an event detection model to capture all of the possible events that occur for communities. This includes tracing the formation, survival and dissolution of communities as well as identifying meta communities, series of similar communities at different snapshots, for any dynamic social network. Additionally, we apply our framework on the Enron Email dataset, and DBLP co-authorship dataset. On these two datasets, our framework uncovers communities with different evolutionary characteristics and addresses the noticeable effect of the similarity threshold on the transitions of communities. We therefore, propose extracting and investigating frequently used topics for each community and select the appropriate similarity threshold based on the continuation of topics. Finding patterns of social interaction within a dynamic network has a wide-range of applications such as disease modeling and information transmission. In our future work, we plan to extend our framework so that it can detect patterns of evolution in dynamic social networks. This includes using the framework to predict the future structure of communities by capturing abstract patterns of evolution in the social network.

References

Airoldi, E. M., Blei, D. M., Fienberg, S. E., & Xing, E. P. (2008). Mixed membership stochastic blockmodels. *Journal of Machine Learning Research*, *9*, 1981–2014.

Asur, S., & Parthasarathy, S. (2009). A viewpoint-based approach for interaction graph analysis. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining* KDD '09 (pp. 79–88).

Asur, S., Parthasarathy, S., & Ucar, D. (2007). An event-based framework for characterizing the evolutionary behavior of interaction graphs. In *Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining* KDD '07 (pp. 913–921).

Backstrom, L., Huttenlocher, D., Kleinberg, J., & Lan, X. (2006). Group formation in large social networks: membership, growth, and evolution. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining* KDD '06 (pp. 44–54).

Berger-Wolf, T. Y., & Saia, J. (2005). *Critical Groups in Dynamic Social Networks.*. DIMACS Technical Report.

Berger-Wolf, T. Y., & Saia, J. (2006). A framework for analysis of dynamic social networks. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining* KDD '06 (pp. 523–528).

Bernstein, A., Clearwater, S., Hill, S., Perlich, C., & Provost, F. (2002). Discovering knowledge from relational data. In *Proceedings of the KDD-2002 Workshop on Multi-Relational Data Mining* MRDM '02 (pp. 7–20).

Calvo-Armengol, A., & Zenou, Y. (2003). *Social Networks and Crime Decisions: The Role of Social Structure in Facilitating Delinquent Behaviour*. CEPR Discussion Papers 3966.

Chakrabarti, D., Kumar, R., & Tomkins, A. (2006). Evolutionary clustering. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining* KDD '06 (pp. 554–560).

Chen, J., Fagnan, J., Goebel, R., Rabbany, R., Sangi, F., Takaffoli, M., Verbeek, E., & Zaïane, O. R. (2010). Meerkat: Community mining with dynamic social networks. In *Proceedings of 10th IEEE International Conference on Data Mining* ICDM '10.

Chen, J., Zaïane, O. R., & Goebel, R. (2009). Local community identification in social networks. In *Proceedings of the International Conference on Advances in Social Networks Analysis and Mining* ASONAM '09.

Choi, D. S., Wolfe, P. J., & Airoldi, E. M. (2010). Stochastic blockmodels with growing number of classes. *CoRR*, *abs/1011.4644*.

Doreian, P., Batagelj, V., & Ferligoj, A. (2004). Generalized blockmodeling of two-mode network data. *Social Networks*, *26*, 29–53.

Doreian, P., Batagelj, V., & Ferligoj, A. (2005). *Generalized Blockmodeling*. Cambridge University Press.

Eubank, S., Guclu, H., Anil Kumar, V. S., Marathe, M. V., Srinivasan, A., Toroczkai, Z., & Wang, N. (2004). Modelling disease outbreaks in realistic urban social networks. *Nature*, *429*, 180–184.

Falkowski, T., Bartelheimer, J., & Spiliopoulou, M. (2006). Mining and visualizing the evolution of subgroups in social networks. In *Proceedings of the 2006 IEEE/WIC/ACM International Conference on Web Intelligence* WI '06 (pp. 52–58).

Falkowski, T., Barth, A., & Spiliopoulou, M. (2008). Studying community dynamics with an incremental graph mining algorithm. In *Proceedings of the 14th Americas Conference on Information Systems* AMCIS '08.

Ganti, V., Gehrke, J., Ramakrishnan, R., & Loh, W.-Y. (2002). A framework for measuring differences in data characteristics. *Journal of Computer and System Sciences*, *64*, 542–578.

Greene, D., Doyle, D., & Cunningham, P. (2010). Tracking the evolution of communities in dynamic social networks. In *Proceedings of International Conference on Advances in Social Networks Analysis and Mining* ASONAM'10.

Handcock, M. S., Raftery, A. E., & Tantrum, J. M. (2007). Model-based clustering for social networks. *Journal of the Royal Statistical Society: Series A (Statistics in Society)*, *170*, 301–354.

Kempe, D., Kleinberg, J., & Tardos, E. (2003). Maximizing the spread of influence through a social network. In *Proceedings of the 9th ACM SIGKDD international conference on Knowledge discovery and data mining* KDD '03 (pp. 137–146).

Kuhn, H. W. (1955). The Hungarian method for the assignment problem. *Naval Research Logistic Quarterly*, *2*, 83–97.

Kumar, R., Novak, J., & Tomkins, A. (2006). Structure and evolution of online social networks. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining* KDD '06 (pp. 611–617).

Leskovec, J., Kleinberg, J., & Faloutsos, C. (2005). Graphs over time: densification laws, shrinking diameters and possible explanations. In *Proceedings of the 11th ACM SIGKDD international conference on Knowledge discovery in data mining* KDD '05 (pp. 177–187).

Lin, Y.-R., Chi, Y., Zhu, S., Sundaram, H., & Tseng, B. L. (2008). Facetnet: a framework for analyzing communities and their evolutions in dynamic networks. In *Proceeding of the 17th international conference on World Wide Web* WWW '08 (pp. 685–694).

Meyers, L., Newman, M., & Pourbohloul, B. (2006). Predicting epidemics on directed contact networks. *Journal of Theoretical Biology*, *240*, 400–418.

Mucha, P. J., Richardson, T., Macon, K., Porter, M. A., & Onnela, J.-P. (2010). Community structure in Time-Dependent, multiscale, and multiplex networks. *Science*, *328*, 876–878.

Newman, M. E. J. (2006). Finding community structure in networks using the eigenvectors of matrices. *Physical Review E*, *74*, 036104.

Newman, M. E. J., & Girvan, M. (2004). Finding and evaluating community structure in networks. *Physical Review E*, *69*, 026113.

Newman, M. E. J., & Park, J. (2003). Why social networks are different from other types of networks. *Physical Review E*, *68*, 36122.

Oliveira, M. C. M., & Gama, J. (2010). Bipartite graphs for monitoring clusters transitions. In *Proceedings of the 9th International Conference on Intelligent Data Analysis* (pp. 114–124).

Palau, J., Montaner, M., López, B., & de la Rosa, J. L. (2004). Collaboration analysis in recommender systems using social networks. In *Proceedings of the 8th International Workshop on Cooperative Information Agents* CIA '04 (pp. 137–151).

Palla, G., Barabasi, A.-L., & Vicsek, T. (2007). Quantifying social group evolution. *Nature*, *446*, 664–667.

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

Sarkar, P., & Moore, A. W. (2005). Dynamic social network analysis using latent space models. *SIGKDD Explor. Newsl.*, *7*, 31–40.

Spiliopoulou, M., Ntoutsi, I., Theodoridis, Y., & Schult, R. (2006). Monic: modeling and monitoring cluster transitions. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining* KDD '06 (pp. 706–711).

Sun, Y., Han, J., Zhao, P., Yin, Z., Cheng, H., & Wu, T. (2009a). Rankclus: Integrating clustering with ranking for heterogeneous information network analysis. In *Proceedings of the 12th International Conference on Extending Database Technology: Advances in Database Technology* EDBT '09 (pp. 565–576).

Sun, Y., Tang, J., Han, J., Gupta, M., & Zhao, B. (2010). Community evolution detection in dynamic heterogeneous information networks. In *Proceedings of the 8th Workshop on Mining and Learning with Graphs* MLG '10 (pp. 137–146).

Sun, Y., Yu, Y., & Han, J. (2009b). Ranking-based clustering of heterogeneous information networks with star network schema. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining* KDD '09 (pp. 797–806).

Takaffoli, M., Sangi, F., Fagnan, J., & Zaïane, O. R. (2010). A framework for analyzing dynamic social networks. In *7th Conference on Applications of Social Network Analysis* ASNA '10.

Takaffoli, M., Sangi, F., Fagnan, J., & Zaïane, O. R. (2011). Modec - modeling and detecting evolutions of communities. In *5th International AAAI Conference on Weblogs and Social Media* ICWSM '11.

Tang, L., Liu, H., Zhang, J., & Nazeri, Z. (2008). Community evolution in dynamic multi-mode networks. In *Proceeding of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining* KDD '08 (pp. 677–685).

Tantipathananandh, C., Berger-Wolf, T. Y., & Kempe, D. (2007). A framework for community identification in dynamic social networks. In *Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining* KDD '07 (pp. 717–726).

Tyler, J. R., Wilkinson, D. M., & Huberman, B. A. (2003). Email as spectroscopy: Automated discovery of community structure within organizations. In *Communities and technologies* (pp. 81–96).

Wasserman, S., & Faust, K. (1994). *Social Network Analysis: Methods and Applications*. Cambridge University Press.

White, H., Boorman, S., & Breiger, R. (1976). Social structure from multiple networks: I. blockmodels of roles and positions. *American Journal of Sociology*, *81*, 730–80.

White, S., & Smyth, P. (2005). A spectral clustering approach to finding communities in graph. In *Proceedings of SIAM International Conference on Data Mining*.

Witten, I. H., Paynter, G. W., Frank, E., Gutwin, C., & Nevill-Manning, C. G. (1999). KEA: Practical automatic keyphrase extraction. In *ACM DL* (pp. 254–255).

Wu, B., Ye, Q., Yang, S., & Wang, B. (2009). Group crm: a new telecom crm framework from social network perspective. In *Proceeding of the 1st ACM international workshop on Complex networks meet information and knowledge management* CNIKM '09 (pp. 3–10).