MODEC — Modeling and Detecting Evolutions of Communities

Mansoureh Takaffoli, Farzad Sangi, Justin Fagnan, Osmar R. Zaïane
Department of Computing Science, University of Alberta
Edmonton, Alberta, Canada
takaffol,fsangi,fagnan,zaiane@ualberta.ca

Abstract

Social network analysis encompasses the study of networked data and examines questions related to structures and patterns that can lead to the understanding of the data and the intrinsic relationships, such as identifying influential nodes, recognizing critical paths, predicting unobserved relationships, discovering communities, etc. All of these analyses, germane to a variety of application domains, are typically done on static information networks; that is, a fixed snapshot of the information network. Yet, a social network changes and understanding the evolution of the network and detecting these changes in the underlying structures is paramount for a multitude of applications. Looking at networks as fixed snapshots misses the opportunity to capture the evolutionary patterns. In this paper, we present a framework for modeling community evolution in social networks by tracking of events related to the life cycle of a community. We illustrate the capabilities of our framework by applying it to real datasets and validate the results using topics extracted from the tracked communities.

Introduction

Social networks, are interconnected records typically represented by a graph that depicts the interactions between individuals or entities. In these networks, each individual is represented by a node, and there is an edge between two nodes if an interaction has occurred, or a relationship exists, between the two individuals during the time. The analysis of these networks is of interest to many fields such as sociology (Wasserman and Faust 1994), epidemiology (Meyers, Newman, and Pourbohloul 2006), criminology (Calvo-Armengol and Zenou 2003), etc. The need to identify communities, which are densely connected individuals that are loosely connected to others (Newman and Girvan 2004), has recently driven attention in the research community.

Most networks are dynamic as they tend to evolve gradually, due to frequent changes in the interaction of their individuals. Also, the communities inside a dynamic network could grow or shrink, and the membership of the individuals shifts regularly (Newman and Park 2003). In these networks, researchers may be interested in the evolution of communities and membership of individuals. One way to model the structural changes in dynamic networks is to convert an evolving network into static graphs at different snapshots. Such dynamic analysis of social networks, especially assessing the evolution of communities, provides insights into understanding the structures of the networks, and detecting a change in the interaction patterns. Leskovec et al. (Leskovec, Kleinberg, and Faloutsos 2005) study the patterns of growth for graphs based on various topological properties, such as the degree of distribution and small-world properties of large networks. Backstrom et al. (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. (Kumar, Novak, and Tomkins 2006) provide the properties of two real-world networks and then analyze the evolution of structure in these networks. Tantipathananandh et al. (Tantipathananandh, Berger-Wolf, and Kempe 2007) formulate the detection of dynamic communities as a graph coloring problem. They provide a heuristic technique that involves greedily matching detected communities at different snapshots. Falkowski et al. (Falkowski, Barth, and Spiliopoulou 2008) discover the evolution of communities by applying clustering on a graph formed by all detected communities at different timepoints.

A number of researchers are working on identifying events that characterize the evolution of communities in dynamic networks. For example, Palla et al. (Palla, Barabasi, and Vicsek 2007), Asur et al. (Asur, Parthasarathy, and Ucar 2007), Takaffoli et al. (Takaffoli et al. 2010), and Greene et al. (Greene, Doyle, and Cunningham 2010) analyze the behavior of network by defining events between communities detected at two consecutive snapshots and characterized each community by a series of events. All of these works focus on analyzing the evolution of communities by using a two-stage approach, where the communities are first detected independently for each snapshot, and then compared to determine the evolution. Another approach is to use evolutionary community mining, where the community mining at a particular time is influenced by the communities detected in a previous time. Adapting community mining in order to consider both current and historic information into the objective of the mining process is also proposed by (Chakrabarti, Kumar, and Tomkins 2006; Asur and Parthasarathy 2009; Sun et al. 2010). However,

none of the previous works cover all of the changes a community may experience during the observation time of a dynamic social network. Thus, we propose the MODEC framework to detect all the events related to the communities where it takes the detected communities at consecutive snapshots as an input and provides a mapping of how each community evolved at each timepoint.

MODEC Framework

We develop the MODEC framework in order to model and detect the evolution of communities obtained at different snapshots in a dynamic social network. Here, the problem of detecting the transition of communities is reduced to identifying events that characterize the changes of the communities across the time of observation. The events are defined in such a way that they can capture all of the changes a community may experience.

Let $1, ..., n$ be the sequence of snapshots under observation. Graph $G = (V, E)$ denotes an aggregate graph representing the dynamic social network where V and E are all the individuals and interactions respectively over an observation time. 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, Barabasi, and Vicsek 2007; Tantipathananandh, Berger-Wolf, and Kempe 2007), the communities at any snapshot can be the result of any static community mining algorithm. Thus, our framework is independent of the community mining algorithm used. The n_i communities 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 represented by (V_i^p, E_i^p). In the literature there are different taxonomies to categorize the changes of clusters or communities that evolve over time (Spiliopoulou et al. 2006; Asur, Parthasarathy, and Ucar 2007; Palla, Barabasi, and Vicsek 2007). To capture the changes that are likely to occur for a community, we consider five events: *form*, *dissolve*, *survive*, *split*, and *merge*. The key concept for the detection of these events is the matching of communities across time. We define matching as the process of finding a map between the communities obtained at a snapshot and the communities at later snapshots, not necessary consecutive. Thus, for a community C_i^p discovered at ith snapshot, we must first find the matching community at later snapshots. Then, based on the existence of the matching community, the events for C_i^p can be detected. Two communities are matched if at least $k\%$ of their members are the same:

Community Match: Let C_i^p and C_j^q be the communities detected at snapshot i and $j \neq i$ respectively. Community C_j^q is a match for C_i^p at jth snapshot if and only if C_j^q is the community with the maximum mutual members for C_i^p and the mutual members are at least $k\%$ of the largest one:

$$\begin{aligned} &\mathrm{match}(C_i^p, j) = C_j^q \quad \text{iff} \\ &C_j^q = \underset{C_j^u \in C_j}{\arg\max} \left\{ \frac{|V_i^p \cap V_j^u|}{\max(|V_i^p|, |V_j^u|)} \right\} \geq k\% \end{aligned} \tag{1}$$

If there is no such $C_j^q \in C_j$, then $\mathrm{match}(C_i^p, j) = \emptyset$.

A community C_i^p at ith snapshot may undergo different transitions at later snapshots. Community C_i^p *splits* at snapshot $j > i$ if it fractures into more than one community with at least k% of their members from C_i^p. Community C_i^p *survives* if there is a community match for it in any $j > i$ snapshot, when there is no community match for C_i^p at later snapshots the community *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 jth snapshot if it fractures to more than one community and one of these communities is the community match for C_i^p; community C_i^p *splits* and *dissolves* at jth snapshot if it fractures to other communities and none of these communities are the community match for C_i^p. A set of communities in C_i can *merge* in Community C_j^q at snapshot $j > i$. The *merge* event occurs when at least $k\%$ of the members from more than one communities in C_i, exist in C_j^q. At any snapshot there may be newly formed communities that are the ones that do not have any match in previous snapshots. The definitions of these events are as follows:

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

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

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

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

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:

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

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 $k\%$ 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, at least $k\%$ of its member must enclosed in C_j^*:

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

Merge: A set of communities $C_i^* = \left\{C_i^1, ..., C_i^n\right\}$ at ith snapshot merges to C_j^q at snapshot $j > i$ if C_j^q contains at least k% 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, at least $k\%$ of its member must enclosed in C_i^*:

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

Experiments

In this section, we validate the feasibility of MODEC through experiments on two real datasets: Enron email dataset and DBLP co-authorship dataset. On both these datasets, we compare the MODEC framework with the other event-based frameworks including Asur et al. (Asur, Parthasarathy, and Ucar 2007), Palla et al. (Palla, Barabasi, and Vicsek 2007), and Greene et al. (Greene, Doyle, and Cunningham 2010) using the automatic extraction and the investigation of the topics of communities. Due to computational efficiency, we apply the local community mining algorithm (Chen, Zaïane, and Goebel 2009) to produce sets of disjoint communities for each snapshot.

The Enron email dataset contains the emails between employees of the Enron Corporation. The dataset includes a period of 15 years, however, the last year (2001) is chosen which results in a graph with 250 nodes. We set the snapshots to be one month each and find the communities on each month by the chosen local community mining algorithm. 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, the similarity threshold k is set to 0.5 and then MODEC finds each community with the appropriate events. In Figure 1 communities at each timeframe are marked with different colours, where these colours are the notion of community match and survival events (the communities without color are the ones that only exist for one snapshot).

To assess the validity of our detected events, we evaluate the topics discussed by the members of these communities and the change of topics in time. 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. Then, the topics for each community correspondent to its 10 most frequent keywords is extracted by KEA. We expect that a community which survives multiple timeframes is most likely to continue discussions on the same topics. Indeed this is the case for the community labeled by A in Figure 1, as Transwestern Pipeline Company, was consistently its most frequently discussed topic for the whole year. We also expect that a community resulting from a *merge* would be discussing a medley of the topics that were present in the previous communities. For example, when community B and C in March merged, the resulting community B continued discussing many topics from B and fewer topics from C: Federal Energy Regulatory Commission, and Pacific Gas and Electric Company, which are the frequent topics of B and C respectively, are also discussed in the merged community. However, the majority of the topics in the merged community are from B, thus confirming the survival event. A similar expectation is also made for *split*. For instance, when the merged community B splits to two communities in May, the resulting B and C discussed the same topics as they did separately before the merger.

The comparison of MODEC with the other frameworks is shown in Table 1, where the total number of events detected by each framework during the 12 snapshots is provided. Applying Asur's framework, only a few *merge*, *split*, *form*, and *dissolve* events are captured. This framework could not

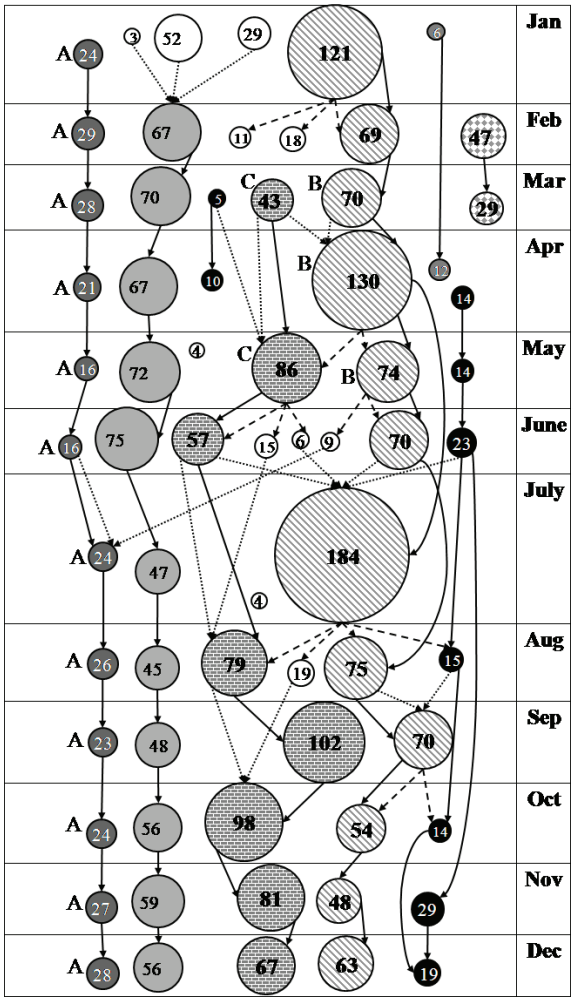

Figure 1: Events detected by the MODEC framework. Solid, dashed, and dotted arrows show detected *survive*, *split*, and *merge*.

detect any *survive* events due to its restricted definition of these events and also because it only considers events between two consecutive snapshots. Palla's framework defines events based on the concept of matching communities across time. However, the framework can not find matches for many communities, thus, no events are detected for these communities. Greene's framework can not discover most of the *merge* and *split* events occurring during the observation time. Also, some of the *survive* events are not detected by this framework which leads to a higher number of *form* and *dissolve* events than MODEC. The question that arises here is which framework results in the most appropriate community evolutions for the Enron dataset. To evaluate the community evolutions, we again incorporate topics extraction for each community. 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 is calculated for each framework (Table 1). The survival communities mostly discuss the same topics, thus, the framework that corresponds to the highest average mutual topics illustrates the transi-

Table 1: Comparison of different frameworks on Enron dataset.

Framework	Form	Dissolve	Survive	Split	Merge	Mutual Topics
MODEC	19	19	47	7	10	2.304/10
Asur	6	5	0	6	7	0/10
Palla	11	9	22	14	17	1.681/10
Greene	24	22	41	0	2	2.048/10

Table 2: Comparison of different frameworks on DBLP dataset.

Framework	Form	Dissolve	Survive	Split	Merge	Mutual Topics
MODEC	2057	2057	576	31	40	14.142/20
Asur	2108	2111	30	18	13	1.966/20
Palla	1336	1242	331	110	122	7.083/20
Greene	2261	2246	356	1	15	7.668/20

tions of the communities better than the others. Our results show that the highest mutual topics out of the top 10 most frequent keywords is found when using MODEC to detect the evolution of communities, thus, MODEC results in the most meaningful community transitions.

Our next dataset is a subset of the DBLP, where the co-authorship network for three major data mining conferences including ICDM, SIGMOD, and KDD from year 2000 to 2009 is chosen. The resulting network contains approximately 7000 individuals, with each year determining one snapshot. In DBLP, communities can be highly dynamic where members leave gradually, while new ones join, thus, a rather low similarity threshold ($k = 0.4$) is chosen. The events detected in DBLP are also validated by extracting the topics from the titles and abstracts of the papers published within communities. The comparison of MODEC with the other frameworks on DBLP is shown in Table 2 where the total number of events detected by each framework during the 10 years and the average mutual topics out of the top 20 most frequent keywords between survival communities is provided. Again, Asur's framework can not detect most of the *survive* events, thus, the number of *form* and *dissolve* events is higher. Applying Palla's framework, many communities remain unmatched so no events are discovered for them. The higher number of *form* and *dissolve* events found by Greene's framework is because of many undetected *survive* events. Again, our results shows that MODEC results in the highest average mutual topics, thus, provides the most meaningful community transitions.

Conclusion

We present MODEC to monitor community evolutions over time, which includes tracing the formation, survival and dissolution of communities in a dynamic social network. Applying MODEC on the Enron dataset, we visualize the events that occurred in Enron Corporation's final year. These events are validated by extracting the topics of emails exchanged within communities, and the performance of MODEC is compared with the other event-based frameworks. We also applied MODEC on a subset of DBLP, and the DBLP events are also validated by extracting the topics of papers published in communities. On both databases, MODEC outperforms the others based on the average mutual topics between survival communities.

References

Asur, S., and Parthasarathy, S. 2009. A viewpoint-based approach for interaction graph analysis. In *KDD*, 79–88.

Asur, S.; Parthasarathy, S.; and Ucar, D. 2007. An event-based framework for characterizing the evolutionary behavior of interaction graphs. In *KDD*, 913–921.

Backstrom, L.; Huttenlocher, D.; Kleinberg, J.; and Lan, X. 2006. Group formation in large social networks: membership, growth, and evolution. In *KDD*, 44–54.

Calvo-Armengol, A., and 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.; and Tomkins, A. 2006. Evolutionary clustering. In *KDD*, 554–560.

Chen, J.; Zaïane, O. R.; and Goebel, R. 2009. Local community identification in social networks. In *ASONAM*.

Falkowski, T.; Barth, A.; and Spiliopoulou, M. 2008. Studying community dynamics with an incremental graph mining algorithm. In *Americas Conference on Information Systems*.

Greene, D.; Doyle, D.; and Cunningham, P. 2010. Tracking the evolution of communities in dynamic social networks. In *ASONAM*.

Kumar, R.; Novak, J.; and Tomkins, A. 2006. Structure and evolution of online social networks. In *KDD*, 611–617.

Leskovec, J.; Kleinberg, J.; and Faloutsos, C. 2005. Graphs over time: densification laws, shrinking diameters and possible explanations. In *KDD*, 177–187.

Meyers, L.; Newman, M.; and Pourbohloul, B. 2006. Predicting epidemics on directed contact networks. *Journal of Theoretical Biology* 240(3):400–418.

Newman, M., and Girvan, M. 2004. Finding and evaluating community structure in networks. *Phys Rev E* 69:026113.

Newman, M., and Park, J. 2003. Why social networks are different from other types of networks. *Phys Rev E* 68:36122.

Palla, G.; Barabasi, A.-L.; and Vicsek, T. 2007. Quantifying social group evolution. *Nature* 446(7136):664–667.

Spiliopoulou, M.; Ntoutsi, I.; Theodoridis, Y.; and Schult, R. 2006. Monic: modeling and monitoring cluster transitions. In *KDD*, 706–711.

Sun, Y.; Tang, J.; Han, J.; Gupta, M.; and Zhao, B. 2010. Community evolution detection in dynamic heterogeneous information networks. In *MLG*, 137–146.

Takaffoli, M.; Sangi, F.; Fagnan, J.; and Zaïane, O. R. 2010. A framework for analyzing dynamic social networks. In *Applications of Social Network Analysis*.

Tantipathananandh, C.; Berger-Wolf, T. Y.; and Kempe, D. 2007. A framework for community identification in dynamic social networks. In *KDD*, 717–726.

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

Witten, I. H.; Paynter, G. W.; Frank, E.; Gutwin, C.; and Nevill-Manning, C. G. 1999. Kea: Practical automatic keyphrase extraction. In *Digital libraries*, 254–255.