Tracking the Evolution in Social Network: Methods and Results

Shengqi Yang, Bin Wu, and Bai Wang

Beijing Key Laboratory of Intelligent Telecommunications Software and Multimedia,
School of Computer Science, Beijing University of Posts and Telecommunications,
Beijing 100876, China
sheng_qi.yang@yahoo.com.cn

Abstract. Contrary to previous static knowledge, our dynamic view in social network is so limited. Recent uncovering those hidden dynamic patterns has posed a series of challenging problems in network evolution. To make effective exploration, we present a fundamentally novel framework for uncovering the intricate properties of evolutionary networks. Different from static snapshots methods, we firstly trace the timelines of networks, which could explicitly characterize the network to several evolving segments. Then based on extracted smooth segments from the timeline, a graph approximation algorithm is devised to capture the frequent characteristics of the network and reduce the noise of interactions. Moreover, by employing the relationship between multi-attributes, an innovative community detection algorithm is proposed for detailed analysis on the approximate graphs. Besides the algorithms, to track these dynamic communities, we also introduce a community correlation and evaluation criterion. Finally, applying this framework to several synthetic and real-world datasets, we demonstrate the critical relationship between event and social evolution, and find that close-knit relationship with well-distributed tie strengths among members of large communities will contribute to a longer life span.

Keywords: Social Network, Dynamic patterns, Community Detection.

1 Introduction

Dynamic properties and evolving patterns have currently caught a considerable amount of attention in complex social networks. Many temporal analytic methods have been actually implemented in various social networks [5][6][11]. In these methods, graphs are usually employed to describe the network during a particular snapshot (e.g., one day, one week or one month) rather than the whole lifespan. Based on these sequential graphs, the temporal properties can be extracted [7][9]. However, these seemingly suitable methods neglect the randomness and emergency of the dynamic interactions, which can be helpful to uncover critical social structures and behaviors (such as criminal gangs). In brief, there are mainly two concernful properties neglected in prior *hard snapshot* methods:

J. Zhou (Ed.): Complex 2009, Part I, LNICST 4, pp. 693–706, 2009.

- Interaction Noise: Interactions of individuals may appear alternately. Such noise may be amplified and affect the final results.
- Event: Traditional evolutionary analysis just neglected the function of events. However, in most real-world networks, event is a driving property of social evolution.

Considering both of these problems, the main contributions in this paper can be induced into four parts: (a)Although previous works [3][5] have promoted the concept of *timeline*, it is just a side product or can not be implemented efficiently. In this work, we propose a efficient method that can generate the timeline throughout the lifecycle of networks without any preconditions, which will also guild our further analysis; (b)Different from traditional graph approximation on single sparse graph [8], we propose a novel approximation method to abstract graph sequence of a network. The goal in graph approximation in our framework is to sculpt naturally occurring structure. This makes it an ideal technique for characterizing the graph segments while avoiding noise; (c)To analyze communities and their evolutions in a unified process with our framework, we elaborate a community detection algorithm based on the weighted approximate graphs. Our method, which is well adjusted to suit our framework, can well reveal the hidden group structures of network and yield good results both efficiently and effectively; (d)A variety of recent works have been mining multi-dimension properties in dynamic networks [1][7][10]. To track the community evolution, we propose a community correlation and evaluation method that can be used practically for uncovering inherent patterns. By applying our method to several real-world networks, we find out that large community with certain compact structure usually survive for a long span.

The rest of this paper is organized as follows: Section 2 presents the notations and definitions used in this paper. Section 3 introduce the datasets that used to verify our framework. In section 4, we describes our framework and algorithms in detail. The corresponding experimental results and statistical analysis are soundly presented. Finally, we conclude our work in Section 5.

2 Notation and Definition

Table 1 lists the basic symbols used throughout this paper. Given a evolving network made up of n snapshots, we describe it as

$$G = \{\mathcal{G}^{(1)}, \mathcal{G}^{(2)}, \dots, \mathcal{G}^{(n)}\} \tag{1}$$

Definition 1 (GRAPH SEGMENT). *A graph segment consisted of* n *snapshots is defined as* $\mathcal{S}^{(i)} = \{\mathcal{G}^{(t)}, \mathcal{G}^{(t+1)}, \dots, \mathcal{G}^{(t+n)}\}(n \geq 0)$. i *is the index of the segment.*

A network can be represented as a series of sequential and non-overlapping graph segments as $G = \{\mathcal{S}^{(1)}, \mathcal{S}^{(2)}, \dots\}$.

Table 1. Symbols

Symbol	Definition
G	Graph that describes the whole network
$\mathcal{G}^{(t)}$	Graph snapshot at t
$\mathcal{S}^{(i)}$	Graph segment at index i
$\mathcal{T}^{(i)}$	Approximate Graph based on $\mathcal{S}^{(i)}$
$\mathcal{C}_j^{(i)}$	Community in $\mathcal{T}^{(i)}$ with index j
$V(\mathcal{G}^{(t)})$	The node set of $\mathcal{G}^{(t)}$
$E(\mathcal{G}^{(t)})$	The edge set of $\mathcal{G}^{(t)}$
$adj_t(v)$	The neighbors of node v at time t
$d_t(v)$	The degree of node v in $\mathcal{G}^{(t)}$
$w_t(v,u)$	The weight of edge (v,u) in $\mathcal{G}^{(t)}$
$\delta(\mathcal{G}^{(t)}, \mathcal{G}^{(t+1)})$ or $\delta(t, t+1)$	The distance between $\mathcal{G}^{(t)}$ and $\mathcal{G}^{(t+1)}$
$\tilde{d}_{t,t+1}(v)$	The distance of node v between snapshot t and $t+1$
$Cor_n(\mathcal{C}_j^{(i)}, \mathcal{C}_k^{(i+1)})$	Node correlation rate between community $\mathcal{C}_j^{(i)}$ and $\mathcal{C}_k^{(i+1)}$)
$Cor_e(\mathcal{C}_j^{(i)}, \mathcal{C}_k^{(i+1)})$	Edge correlation rate between community $\mathcal{C}_j^{(i)}$ and $\mathcal{C}_k^{(i+1)}$)

Definition 2 (APPROXIMATE GRAPH). *An approximate graph $\mathcal{T}^{(i)}$ is an abstract image of a graph segment $\mathcal{S}^{(i)}$ that can characterize the graph sequence of the segment.*

3 Datasets

Here we collect several datasets to demonstrate our algorithm and suggest its generality in solving problems with complex relationships.

Table 2. Datasets

name	N	E	$\overline{N_t}$	$\overline{E_t}$	$\overline{\rho}$	time span
Random	10k	872k	2k	8.7k	4.36	100
BA 1	10k	877k	1.5k	4.6k	3.0	100
BA 2	10k	890k	1.5k	4.6k	3.0	100
BA 3	10k	890k	1.44k	6.6k	4.6	100
VAST [1]	400	9834	373	983	2.64	10(d)
Enron [2]	150	24k	60	219	3.34	111(w)
cond-mat [3]	52k	280k	1k	4k	3.95	117(m)
Cell Calls A	265	113k	167	812	4.83	118(d)
Cell Calls B	352	54k	196	436	2.23	102(d)
Cell Calls C	64k	1,090k	7.4k	10.8k	1.5	101(d)

[1] From http://www.cs.umd.edu/hcil/VASTchallenge08/
[2] From http://www.cs.cum.edu/enron/
[3] From http://arxiv.org/archive/cond-mat

Since typical real-world data does not have the ground truth (knowledge of changes in structure) available, we resort to ***synthetic datasets*** as a proof of concept to show the efficiency of our algorithm. As in table 2, the four synthetic datasets are built up from two models. *Random* comes from random construction (simple ER model) while *BA 1, 2 and 3* are constructed to comply with BA model. During their evolvement, we update 40% nodes between every two snapshots. *BA 2 and 3* also introduce some events with 2% and 5% high degree nodes disposal respectively to simulate structure changes during the process.

Currently increasing concentration on social behavior patterns of human, including from phone calls to e-mails, often offering particular avenues to explore both static and evolving social structures. In this paper, we also elaborate our framework to uncover such undergoing patterns in real world. Here we would like introduce our collected datasets including both public benchmarks and anonymous mobile call records.

The VAST Dataset is a challenge task from *IEEE VAST 2008*. It describes a set of cell calls from a fictitious island over a ten-day period which was narrowed down to about 400 unique cell phones during this period.

Co-authorship Dataset comes from Cornell e-Print cond-mat library spanning 70 months from 03/2001 to 12/2006. In this dataset, each record stands for a co-authorship experience between two authors.

Enron Email Dataset contains data from about 150 users, mostly senior management of Enron and spans 111 weeks from 12/1999 to 03/2002. During this period, several symbolic events happened, including the collapse.

Company Calls A B are the call records from the same company during different periods. Calls A spans 187 days from 10/2005 to 3/2006, while Calls B spans 152 days from 12/2007 to 4/2008. During B period, there was a change in top management of the company.

Cell Calls C is the cell call records of one province of an operator in China from 12/2007 to 04/2008. The detailed information of each call pair, include the duration and the frequency, is also reserved.

The call data used here is obtained from a mobile service operator. Although a single call communication may not carry much information that can reflect all the aspects of the relationship of the two involved individuals, reciprocal calls of long duration can usually explain a profile of them as a signature of some work-, family-, leisure-, ore service-based relationship [4]. Here, we utilize these mobile call datasets as a proxy of real-world communication network.

We would like to mention that for the purpose of keeping the privacy of each customer, all the phone numbers are identified by a surrogate key so that it is not possible to recover the actual customer.

4 Methods, Algorithms and Experiments

As mentioned in *Introduction*, the framework in this paper divides the exploration process into several steps which is applicable to track the network evolution in multi-dimension. Our primary concern in this section in developing the framework that we generate to solve problems is to make sure that the methods and algorithms are more practical and suitable.

4.1 TimeLine Detection

Although previous works [3][5] have promoted the concept of *timeline*, it is just a side product or can not be implemented efficiently (mainly due to their incremental methods based on ready-made community structures). In our framework, however, we propose a efficient method that can generate the timeline throughout the lifecycle of networks without any preconditions, which will also guild our further steps.

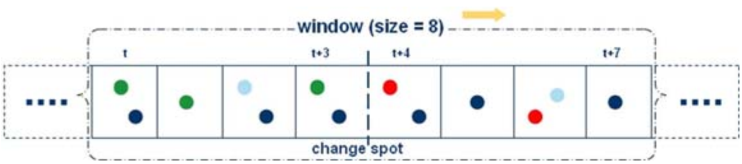


Fig. 1. moving window with size = 8. Green node stands for the nodes dead at change spot. Red node is new born nodes at change spot. Dark blue stands for stable nodes that both appears before and after the change spot while light blue for flush nodes that only appear in one snapshot.

Our algorithm will calculate the value of each change spot illustrated as Fig. 1. Our original method to quantify such change is inspired by the information theories of *relative entropy* and *type method*. The distance $\delta(t, t+1)$ at the change spot is the accumulation of the distance between each corresponding nodes pair $\tilde{d}_{t,t+1}(v)$ in the two graphs, which is defined as:

$$\tilde{d}_{t,t+1}(v) = \begin{cases} |\log \frac{d_t(v)+1}{1}| & v \in \{dead\ nodes\} \\ |\log \frac{1}{d_{t+1}(v)+1}| & v \in \{born\ nodes\} \\ |\log \frac{d_t(v)}{d_{t+1}(v)}| + |\log \frac{adj_t(v) \bigcap adj_{t+1}(v)}{adj_t(v) \bigcup adj_{t+1}(v)}| & v \in \{stablenodes\} \end{cases} \quad (2)$$

In our method, flush nodes are just neglected due to their small proportion[4] and limited effect. For dead or born nodes, we simple focus on their change in degree as Formula 2. While for stable nodes, we depict their change both in degree (their activeness) and neighbors (their environment). The integrated distance is formulated as

$$\delta(t, t+1) = \frac{\sum\limits_{\forall v \in V(dead)} \tilde{d}_{t,t+1}(v) + \sum\limits_{\forall v \in V(born)} \tilde{d}_{t,t+1}(v) + \sum\limits_{\forall v \in V(stable)} \tilde{d}_{t,t+1}(v)}{|V(\mathcal{G}^{(t)}) \cup V(\mathcal{G}^{(t+1)})|} \quad (3)$$

To avoid the side-effect brought about by graph size, here the distance is penalized by the node union of graphs. Then given an evolving graph G, we can get the raw timeline by Formula (3).

[4] The proportion of flush nodes is dependent on the size of window.

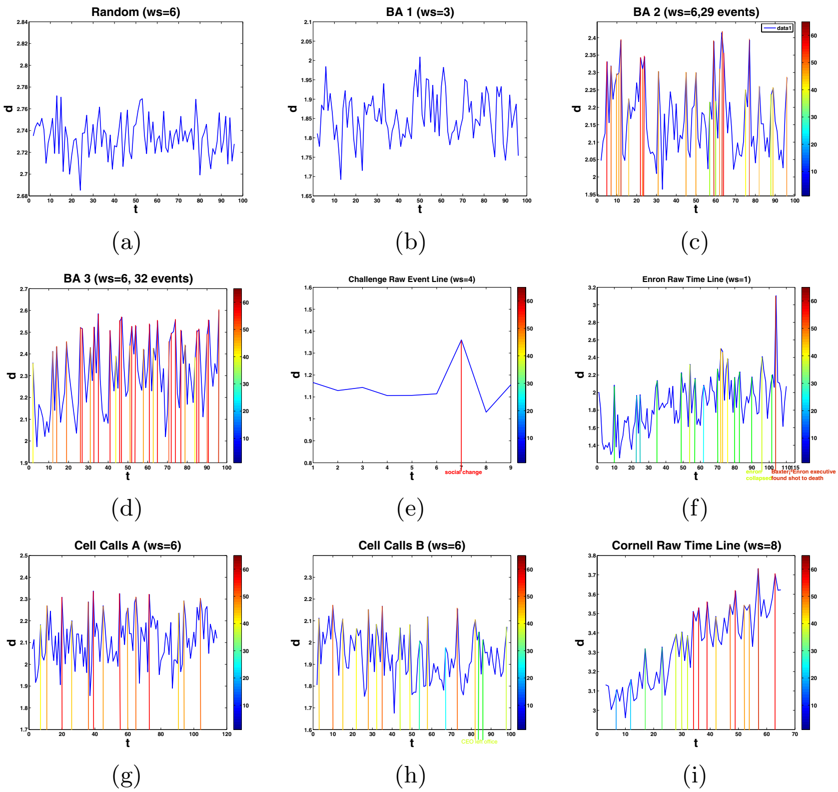


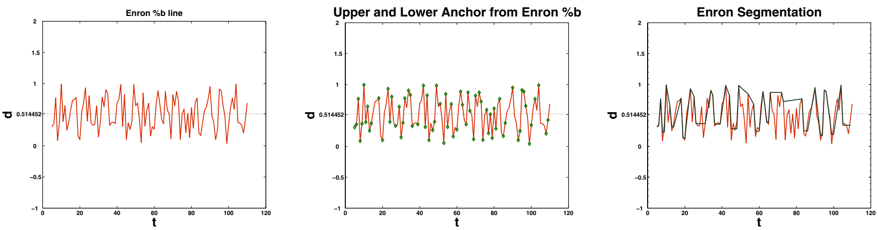
Fig. 2. TimeLines

By applying our algorithm to several synthetic and real-world datasets, the generated timelines are plotted as Fig. 2. Considering the randomness in generation, the timelines of synthetic datasets (a)-(d) are more chaotic than the other 5 real-world[5] datasets (e)-(f). From the perspective of effect, our algorithm locates 24 out of 29 events in BA 2 as marked in (c) and 31 out of 32 events in BA 3 as marked in (d). In real-world, we efficiently find out the critical social emergence in VAST 10 days call records as described in (e) and all the critical events such as company collapse and CEO committed suicide in Enron as marked in (f).

4.2 Graph Segmentation and Approximation

The most recent work which is closely related to mine [5] also segments graph stream in an incremental manner. However, minor deviation could accumulate and form a big distance between graphs, which is also called *butterfly effect*. In

[5] Although VAST dataset is also a synthetic one, it is considered to incorporate many social properties in order to simulate such social behaviors. In this article, we take it as a real one.



(a) (b) (c)

Fig. 3. Segmentation

Table 3. Datasets

	Enron	cond-mat	Cell Calls A	Cell Calls B	Cell Calls C
$\lvert\mathcal{S}^{(i)}\rvert$	13	15	18	18	16
$\overline{\mathcal{S}^{(i)}}$	4	4	5	4	5

general, a network and the membership of groups often evolve gradually [1]. From the social perspective, the evolving network usually alternates between change and smoothness. So the change between two successive smooth segments can be applied to describe the influence of that event. In our framework, we highlight the smooth segments and abstract it into single graphs for farther research.

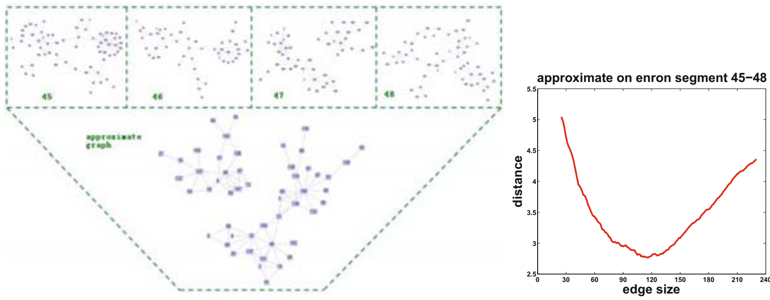
To obtain the smooth segments, here we propose a two-step auto segmentation algorithm to extract smooth graph segments as follows:

1. We employ *Bolling Bands*[14], $\%b$, to prune the raw event line and generate the first base for linear segmentation, which is normalized between 0 and 1. Fig. 3(a) shows the resulted $\%b$ line from Fig. 2(f)
2. The first windowing process is implemented to locate all the upper and lower anchors. The window size is determined by real dataset. Here we apply this process to (a) with window size = 10 and locate all the anchors described as (b). The second windowing process is to smooth the lower anchors and prune upper anchors. The obtained segment line is plotted as black line in (c).

The results by applying the segmentation algorithem to our datasets is summarized as Table 3.

When it comes from segmentation to approximation, we would like to present our objects firstly. (a)The abstraction from $\mathcal{S}^{(i)}$ to a single graph $\mathcal{T}^{(i)}$ should simplify the future evolutionary analysis. (b)$\mathcal{T}^{(i)}$ should well characterize $\mathcal{S}^{(i)}$.

It is intuitively occurred to us that $\mathcal{T}^{(i)}$ should contain only the common structure of a smooth segment. However, it is somewhat over-fit. Our approximation process aims to depict more frequent structures rather then the most.



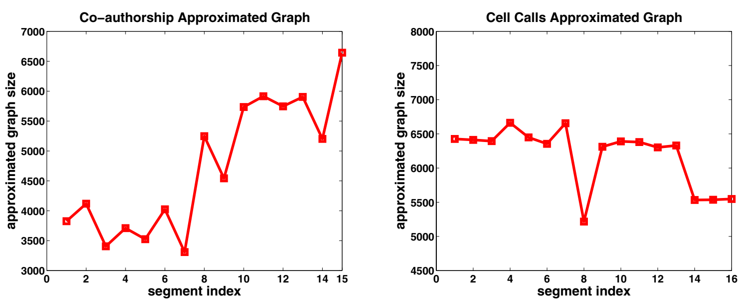
(a) (b)

Fig. 4. Approximation

Then how to obtain a proper approximate graph from the smooth segment? Here we introduce the *Graph Segment Description Distance* between $\mathcal{T}^{(i)}$ and $\mathcal{S}^{(i)}$ as,

$$d(\mathcal{T}^{(i)}, \mathcal{S}^{(i)}) = \sqrt{\sum_{\mathcal{G}^{(t)} \in \mathcal{S}^{(i)}} w_i \delta(\mathcal{T}^{(i)} || \mathcal{G}^{(t)})^2} \tag{4}$$

Even in a smooth segment $\mathcal{S}^{(i)}$, there are also some small events among its snapshots. We find out the lowest point, $\delta(t, t+1)$, in $\mathcal{S}^{(i)}$ and set $w_t = 1$ and $w_{t+1} = 1$. Other weights can be set by $\frac{\delta(j,j+1)}{\delta(t,t+1)}$. Along the adding procedure, we can expect that, at the beginning, when the edges with high weight and frequency are added to $\mathcal{T}^{(i)}$, the value of the $d(\mathcal{T}^{(i)}, \mathcal{S}^{(i)})$ will conspicuously decrease. However when it comes to the low ranked edges, the distance increases reversely. To demonstrate our method, we extract the segment $\mathcal{S}^{(45-48)}$ (include 4 graph snapshots) from Fig. 3(c) and by employing our algorithm, the curve of $d(\mathcal{T}^{(i)}, \mathcal{S}^{(i)})$ which describes the process of approximation on $\mathcal{S}^{(45-48)}$ shows the shape V in Fig. 4(d). Fig 4(a) shows the approximate graph.



(a) (b)

Fig. 5. Approximation on cond-mat and cell calls C

The generated graphs are usually tight relevant with their correspond timelines, which can be observed from Fig. 5. In Fig. 5(a). It is clearly that graph size is almost as the rising shape as the timeline in Fig 2(i). By comparison, the size of approximate graphs in *Cell Calls C* (Fig. 5(d)) is relatively smooth.

Besides precisely depicting segment, as the same time the event or certain great change between two $\mathcal{T}^{(i)}$ can also be observed, which can be employed to explore the social transformation. Fig. 6(a) and (b) describe the change effect at the 7^{th} day in Fig. 2(e). It is clear that there is a significant event happened at the high-level leaders. The replacement, such as from node 200 to 300, who are recognized as the leaders, is obvious before and after the event. In Fig. 6(c)

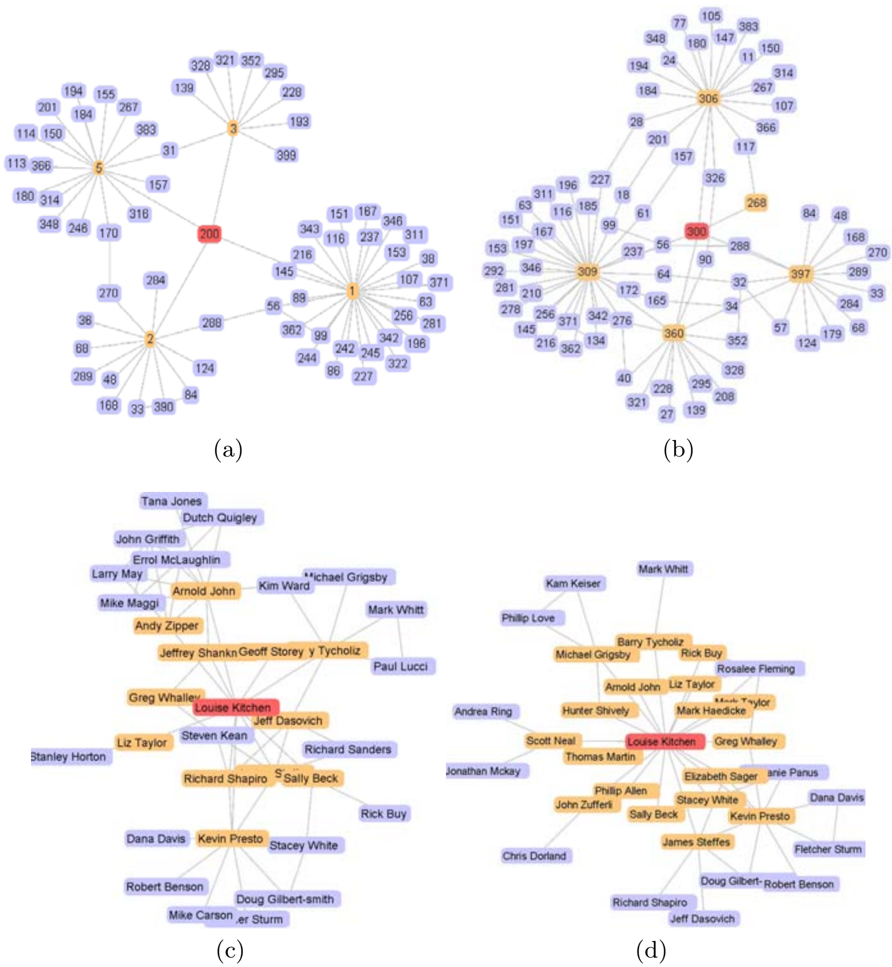


Fig. 6. Graph Tracking: (a) and (b) picture the main structure of VAST before and after the event; (c) and (d) picture the main structure of Enron before and after the company collapse (*Louise Kitchen* labeled in red was the president of Enron)

and (d), we extract the main top-level structures of Enron before and after the event around time 96 in Fig. 2(f), when the company filed for bankruptcy protection (2001/12/2). Fig. 6(c) describes the structure before the event. Most of the people around *Louise Kitchen* are *vise president*. However in Fig. 6(d), there are also some other people whose position may reflect the event, such as *Rick Buy* (the *Chief Risk Management Officer*) and *Mark Haedicke* (a managing director of *Legal Department*).

4.3 Community Tracking

Community Detection. Although traditional modularity-based optimal methods [13][15] yielded a good result on static graphs, they are not applicable to multi-attribute graphs. In this subsection, we elaborate a community detection algorithm based on approximate graphs. A rough description of our algorithm is presented as:

1. Given an approximate graph $\mathcal{T}^{(i)}$, we extract all k-clique communities [12] where k is pre-determined by actual networks.
2. We split the overlapping communities by the weight, $w_i(v, \mathcal{C}_j^{(i)})$, of interactions between v and its adjacent communities $\{\mathcal{C}_j^{(i)}, \mathcal{C}_k^{(i)}, \ldots\}$. By finding out the maximum value, the node is adjudged to that community.
3. In this step, each community try to attract the peripheral nodes around it with the nodes judgement threshold Q_v. That is to say, for any of the adjacent communities of v, v can be adjudged to $\mathcal{C}_j^{(i)}$ only if $w_i(v, \mathcal{C}_j^{(i)}) > Q_v$.
4. In the last step, communities that are tightly connected are merged. A user-defined threshold Q_c is compared with $w_i(\mathcal{C}_j^{(i)}, \mathcal{C}_k^{(i)})$. $\mathcal{C}_j^{(i)}$ and $\mathcal{C}_k^{(i)}$ can be merged if $w_i(\mathcal{C}_j^{(i)}, \mathcal{C}_k^{(i)})$ is greater than Q_c.

We should mention that k, Q_v and Q_c are all determined by actual networks. Higher values will result in more compact communities and fewer nodes coverage.

From the perspective of efficiency, by applying our Clique-Based Community Detection algorithm (*CBCD* for short) to several datasets, table 4 illustrates the results on 4 aspects. For our concentration primarily on more representative structure of dynamic graphs, the communities resulted from *CBCD* are not obligated to cover all the nodes. So the number and size of the communities extracted by our method is relatively smaller than the other two algorithms. However, from the efficiency aspect, *CBCD* which is comparable to *FAST* have an obvious advantage over *GN*. Because of the broad consensus that dramatic change in a short time is unlikely [1][3], plausible community partition methods should result in lower *community fluctuation* (Flu_c) or higher *community correlation* (Cor_c). From the last column of table 4, *CBCD* produces much higher Cor_c than the others.

Community Correlation and Evaluation. Community tracking is a continuous topic in evolutionary analysis. Traditional methods in earlier works mainly

Table 4. Results of community detection algorithms on call graphs

name	$\lvert\{\mathcal{T}^{(i)}\}\rvert$	Alg.	avg. Comm. N.	avg. Comm. S.	avg. T.[7]	avg. Cor_c
Cell Calls A	18	GN[15]	12.9	12.9	4.3s	0.019
		FAST[13]	10.7	15.7	0.10s	0.015
		CBCD[6]	9.6	12.7	0.46s	0.037
Cell Calls B	19	GN	14	11.1	1.5s	0.016
		FAST	22.6	7.9	0.13s	0.015
		CBCD	6.3	10.1	0.35s	0.017
Cell Calls C	16	GN	356	9.5	59.3s	0.031
		FAST	221.1	12.6	1.8s	0.031
		CBCD	106	8.8	1.9s	0.037

focused on either node overlapping [9] or structure (edge) overlapping [7]. However, these two criterions have apparently weak points. In our framework, we take both these two correlations into account.

To evaluate the evolutionary trend of communities, we propose a *community fluctuation criterion*. For graph $\mathcal{G}^{(i)}$ and $\mathcal{G}^{(i+1)}$, which respectively contain m and n communities, the community correlation is formulated as

$$Cor_c(\mathcal{G}^{(i)},\mathcal{G}^{(i+1)}) = \sum_{\substack{\mathcal{C}_j^{(i)} \\ 1\leqslant j\leqslant m}} \Big(\frac{|N(\mathcal{C}_j^{(i)})|}{|N(\mathcal{G}^{(i)})|} \sum_{\substack{\mathcal{C}_k^{(i+1)} \\ 1\leqslant k\leqslant n}} Cor_n(\mathcal{C}_j^{(i)},\mathcal{C}_k^{(i+1)})Cor_e(\mathcal{C}_j^{(i)},\mathcal{C}_k^{(i+1)})\Big) \quad (5)$$

then the community fluctuation, $Flu_c(\mathcal{G}^{(i)},\mathcal{G}^{(i+1)})$, is defined as

$$Flu_c(\mathcal{G}^{(i)},\mathcal{G}^{(i+1)}) = 1 - Cor_c(\mathcal{G}^{(i)},\mathcal{G}^{(i+1)}) \quad (6)$$

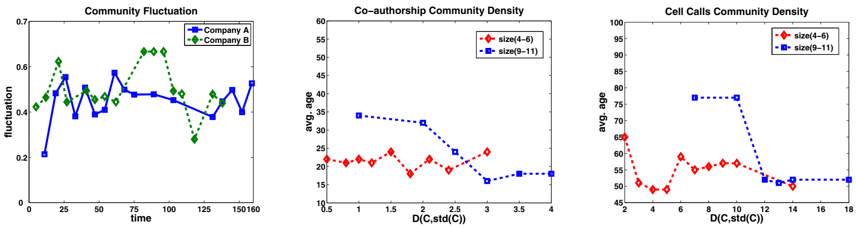
Palla et al. [7] defined the stationarity of a community to evaluate the relationship between its age and the average correlation between subsequent states. However they neglect the effect of inherent structure of communities along the evolvment. To relate the structure of community with its age and evolving correlation, we define the *compactness* of community as the distance between a community and its *standard complete structure* $std(\mathcal{C}_j^{(i)})$, which is not only a complete weighted graph with the same nodes and total edge weight with $\mathcal{C}_j^{(i)}$ but is the most compact structure. The quantity of compactness of $\mathcal{C}_j^{(i)}$ is formulated as

$$Comp(\mathcal{C}_j^{(i)}) = \sum_{\forall(v,u)\in E(std(\mathcal{C}_j^{(i)}))} \Big|\log\frac{w_{\mathcal{C}_j^{(i)}}(v,u)}{w_{std(\mathcal{C}_j^{(i)})}(v,u)}\Big| / |E(\mathcal{C}_j^{(i)})| \quad (7)$$

This representation takes into account both link compactness and weight distribution of the structure. We would like to mention here that a small value of $Comp(\mathcal{C}_j^{(i)})$ stands for a more compact community structure.

[6] With parameters $k = 3, Q_v = 0.6, Q_c = 0.8$.

[7] Intel Xeon CPU 2.60GHz×2, 2G memory.



(a) (b) (c)

Fig. 7. Community Tracking

Discussion. In reality in social networks, timelines are usually well-distributed. That is to say, sometimes, changes only stand for the randomness of interactions but not the events in infrastructure. A negative effect brought about in this situation is that some significant events happened locally may be submerged by the noise of peripheral random interactions. To find the needles in the hay stack, here we employ our community fluctuation method to community tracking. It is obvious that in Fig. 7(a) the line depicting the community fluctuation along the evolution of Cell Calls A (solid blue line) has a relative smooth line. But for Calls B (dashed green line), there is a quite high valued segment between 80 and 100, which implies there is a significant fluctuation in the community structure. Actually, during this period, this company has a series of changes in personnel which include many high-level replacements.

From the evolving aspect, it is recently a central issue in analyzing evolving communities. Previous works [2][7][10][11] have already promoted a broad range of seminal properties in community evolution mainly in macroscopic view. In order to uncovering hidden structure properties, we employ two massive datasets, *cond-mat* and *Cell Calls* to explore their community structure and evolution trend. By imposing our *community evaluation* method to *cond-mat* and *Cell Calls C*, Fig. 7(b) and (c) represent the relations between the average age span at a given community size and the pre-defined compactness $Comp(\mathcal{C}^{(i)})$. In these two statistical plots, we get similar results. To small communities as the red lines describe, the average age span does not vary greatly along the decreasing of the compactness of their structure. By contrast, to larger communities as the blue lines describing, the average age span firstly maintains a relative high value, then decreases sharply within a limited range and at last tends to be stable.

Our insight into the evolving community structure shows that a close-knit relationship with well-distributed link weight among members of large communities will contribute to a longer life span while the phenomenon is not marked in small communities.

5 Conclusion

In this paper, we aim to uncover the evolving patterns and temporal behaviors related with social changes. In contrast with hard snapshot extraction of the evolving network, we employ timeline to segment networks. By approximating the smooth segments, the generated graphs characterize the graph sequence of smoothly evolving segments. This two-step method effectively avoids noise and is more reasonable to reflect the infrastructure of these segments. The following experiments present the event effect on the structure of those social networks. For community exploration, we find that, in Co-authorship and Cell Calls, the community structure is closely relevant to its age span.

In the future, we will improve our timeline detection method to be more effective. Our exclusive community detection is somewhat arbitrary in reality. A next step of our work is to analyze the activities of core nodes or overlapping nodes along the evolution. There are more factors, which could affect the temporal behaviors of the community or even the whole network, to be explored.

References

1. Tang, L., Liu, H., Zhang, J.P., Nazeri, Z.: Community Evolution in Dyanmic Multi-Mode Networks. In: Proceeding of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining, pp. 677–685 (2008)
2. Leskovec, J., Backstrom, L., Kumar, R., Tomkins, A.: Microscopic Evolution of Social Networks. In: Proceeding of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining, pp. 462–470 (2008)
3. Lin, Y.R., Chi, Y., Zhu, S.H., Sundaram, H., Tseng, B.L.: Facetnet: a framework for analyzing communities and their evolutions in dynamic networks. In: Proceeding of the 17th international conference on World Wide Web, pp. 685–694 (2008)
4. Onnela, J.P., Saramaki, J., Hyvonen, J., Szabo, G., Lazer, D., Kaski, K., Kertesz, J., Barabasi, A.L.: Structure and tie strengths in mobile communication networks. In: Proceedings of the National Academy of Sciences, vol. 104, pp. 7332–7336 (2007)
5. Sun, J., Faloutsos, C., Papadimitriou, S., Yu, P.S.: GraphScope: Parameter-free Mining of Large Time-evolving Graphs. In: Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining, pp. 687–696 (2007)
6. Asur, S., Parthasarathy, S., Ucar, D.: 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, pp. 913–921 (2007)
7. Palla, G., Barabasi, A.-L., Vicsek, T.: Quantifying social group evolution. Nature 446, 664–667 (2007)
8. Long, B., Xu, X.Y., Zhang, Z.F., Yu, P.S.: Community Learning by Graph Approximation. In: Proceedings of the 2007 Seventh IEEE International Conference on Data Mining, pp. 232–241 (2007)
9. Tanya, Y., Berger-Wolf, Jared, Saia.: A Framework for Analysis of Dynamic Social Networks. In: Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining, pp. 523–528 (2006)
10. Backstrom, L., Huttenlocher, D., Kleinberg, J., Lan, X.Y.: 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, pp. 44–54 (2006)

11. Kumar, R., Novak, J., Tomkins, A.: Structure and evolution of online social networks. In: Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining, pp. 611–617 (2006)
12. Derényi, I., Palla, G., Vicsek, T.: Clique percolation in random networks. Phys. Rev. Lett. 29, 94(16), 160–202 (2005)
13. Newman, M.E.J.: Fast algorithm for detecting community structure in networks. Phys. Rev. E 69(6), 066133 (2004)
14. Bollinger, J.A.: Bollinger on Bollinger Bands, 1st edn. McGraw-Hill, New York (2001)
15. Girvan, M., Newman, M.E.J.: Community structure in social and biological networks. Proceedings of the National Academy of Sciences 99(12), 7821–7826 (2002)