Incremental Local Community Identification in Dynamic Social Networks

Mansoureh Takaffoli, Reihaneh Rabbany, Osmar R. Zaïane
Department of Computing Science, University of Alberta,
Edmonton, Alberta, Canada T6G 2E8
Email:{takaffol,rabbanyk,zaiane}@ualberta.ca

***Abstract*— Social networks are usually drawn from the interactions between individuals, and therefore are temporal and dynamic in essence. Examining how the structure of these networks changes over time provides insights into their evolution patterns, factors that trigger the changes, and ultimately predict the future structure of these networks. One of the key structural characteristics of networks is their community structure –groups of densely interconnected nodes. Communities in a dynamic social network span over periods of time and are affected by changes in the underlying population, i.e. they have fluctuating members and can grow and shrink over time. In this paper, we introduce a new *incremental community mining* approach, in which communities in the current time are obtained based on the communities from the past time frame. Compared to previous independent approaches, this incremental approach is more effective at detecting stable communities over time. Extensive experimental studies on real datasets, demonstrate the applicability, effectiveness, and soundness of our proposed framework.**

I. Introduction

A social network shows the structure of relationships between individuals. These relationships are usually defined based on some type of interaction, hence are temporal and changing over time; examples are friendships between people, co-authorship between scholars, email interactions between employees within an organization. One can aggregate all the interactions over time, into one snapshot, to model the network using a static social network. However, by discarding these temporal information, one is not able to detect invaluable evolutionary patterns that are happening inside the network. A better modeling for such a temporal/dynamic social network, is using a sequence of consecutive static snapshots. In this model, each snapshot incorporates interactions that happened in its particular time-frame, the length of which can be determined based on how dynamic is the network. Modeling a dynamic network in this way, enables the study of its structure over time, the detection of how the network evolves, and ultimately the prediction of the future structure of the network .

Communities in dynamic social networks usually have fluctuating members and could grow and shrink over time [1]. Analyzing the evolution of these communities is useful in many applications such as targeted marketing and advertising. The 2010 Edelman Trust Barometer Report [2] shows that 44% of the users respond to the online marketing if there are users in their peer group who have responded to the advertisements. Two main approaches have been followed to study the evolution of communities in a dynamic scenario. In the *independent community mining* approach, the communities at each snapshot are mined independently without considering the temporal information and their relationship to communities at the previous snapshots. Hence, this approach is suitable for social networks with unstable community structures. On the other hand, the *incremental community mining* approach uses the temporal information directly during the detection, where the community mining at a particular time is dependent on the communities detected in the previous timeframe. This approach finds a sequence of communities with temporal similarity and hence, is only suitable for networks with community structures that are stable over time.

In our previous works [3], [4], [5], we have explored the independent community mining and proposed an effective framework for tracking the evolution of communities over time. In this paper, we propose an incremental community mining approach that mines communities by considering the communities discovered at previous times. This conditional community mining approach is more appropriate for tracking more stable communities compared to our previous independent method. The main contribution of our paper is the adoption of the static L-metric approach [6], to compute dynamic communities; where community mining at each snapshot starts by the communities found at the previous snapshot. The communities found at different snapshots, are then matched based on their similarity, and grouped as the instances of the evolving communities over time. Furthermore, to capture the changes that are likely to occur for a dynamic community, we apply our event detection framework [5] to detect critical events (i.e. survive, dissolve, split, merge, form) that characterize the evolution of communities.

II. Related Works and Backgrounds

In the independent community mining, after computing communities for each snapshot, the communities are tracked and matched based on their similarity. Communities at different snapshots that are detected as matches, represent the instances of the same community which spans over time. The intuitive method is to compare two communities of the consecutive time steps with rules based on the size of their intersection. These rules can be used conjointly with the community mining algorithm [7], or heuristic algorithm to match communities based on their interaction [8], [3], or even simplified by tracking specific core nodes that are more representative of their community than others [9]. Although, most of the independent community mining approaches consider matching communities between two consecutive snapshots, a community may not necessarily be observed at consecutive snapshots – it may be missing from one or more intermediate steps. To support these cases, this approach can be extended to consider matching communities at current snapshot to communities at all previous snapshots based on their intersections and time of occurrence [5], [10].

The incremental community mining methods can be classified into cost function methods, and direct methods. *Cost function methods*, first introduced by Chakrabarti et al. [11], try to find communities in a particular snapshot that are meaningful communities of the interactions that exist in that snapshot, and at the same time, are similar to the communities detected at its previous snapshot. These methods consider the former as the snapshot quality and the latter as the history quality, and minimize a cost function which is defined as a trade-off between these two qualities. Lin et al. [12], calculate the snapshot cost, as the KL-divergence between the discovered community structure and the graph observed at that snapshot. Similarly, the temporal cost is defined as the KL-divergence between the communities discovered at the current and previous time. They have introduced the FacetNet framework which extends the overlapping community mining proposed by Yu et al. [13] for static graphs to be applicable for dynamic networks. At each snapshot, the community structure is expressed by the mixture model proposed in [13], and the cost function is used to regularize the community structure at current time based on the community structure at the previous snapshot. Tantipathananandh and Berger-Wolf [14], on the other hand, introduced a cost function that consists of three parts: 1) cost of a node that changes its community affiliation between two snapshots; 2) cost of two nodes belonging to the same community but do not interact 3) cost of two nodes belonging to different communities but do interact. Then, they propose the network community interpretation framework to find a set of communities at each snapshot that minimizes the above three costs and devise an approximation algorithm. The main challenges of the cost function methods is to derive the approximation algorithm and the optimum communities that minimize the cost.

While the cost function methods focus to optimize a new quality measure which incorporates deviation from history, the *direct methods* mine communities at the current snapshot incrementally by considering the communities discovered at the previous time and updateding the community structure as the new data arrives. For example, in the incremental community mining algorithm based on the Dirichlet Process Mixture Model, the discovered communities at the previous snapshot are included in the base distribution of the Dirichlet Process [15]. Aggarwal and Yu [16] propose to generate the differential graph between the graphs at the current and previous snapshots. The communities at the differential graph are then mined using k-mean community mining. Kim and Han [17] propose to recalculate the weight between each pair of nodes to be a linear combination of their weights in the previous and current snapshots. Density based clustering is then applied with the new balanced weights to detect communities. Aynaud, and Guillaume [18] extend the Louvain static community mining [19], where the initialization of the Louvain algorithm is changed to use the communities found at the previous snapshot.

In this paper, we extend the L-metric community mining algorithm by Chen et al. [6], to incrementally mine communities in a dynamic scenario following the direct approach. Similar to [18], the main idea here is to change the initialization of the algorithm in a way that the computation at each snapshot starts by grouping nodes using the communities found at the previous snapshot. Our proposed method, however, benefits form the locality of the L-metric, which unlike most of static community mining algorithms that implicitly assume global information is always available, detects communities with only local information. This locality makes L-metrics and consequently our method, particularly desirable in case of large real world networks, where the whole graph is usually unavailable. In the following section, the L-metric community mining algorithm is explained in more details.

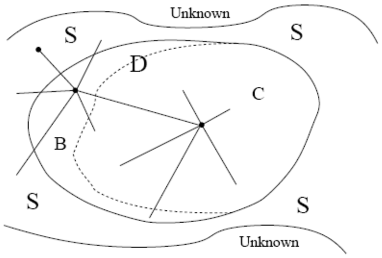


Fig. 1: Local Community Definition. Figure reprinted from [6].

A. L-metric Community Mining

The L-metric does not require any arbitrary thresholds or other parameters, and is robust against outliers. Its main assumption is that a community has fewer connections from its boundary nodes to the unknown portion of the graph, while having a greater number of connections within its local community [6]. In more details, consider an undirected network G, with the known local portion of the graph denote as D. Two subsets of D are defined: the core node set C, where all neighbours of $v \in C$ belong to D; and the boundary node set B, where any node $v \in B$ have at least one neighbour outside D. The shell node set S is the set of nodes with limited available information and contains nodes that are adjacent to nodes in D but do not belong to D (See Figure 1). Then the metric L is defined as the ratio of the community internal relation to the community external relation, i.e. L_{in}/L_{ex}. Where L_{in} is measured by the average internal degree of nodes in D, and L_{ex} is measured by the average external degree of nodes in B. There are three situation in which the modularity L increases after adding one node to the local community. Assume L'_{in}, L'_{ex} and L' are corresponding scores after merging a node v into D. The three cases that will result in $L' > L$ are: 1) $L'_{in} > L_{in}$ and $L'_{ex} < L_{ex}$, 2) $L'_{in} < L_{in}$ and $L'_{ex} < L_{ex}$, and 3) $L'_{in} > L_{in}$ and $L'_{ex} > L_{ex}$ Nodes in the first case belong to the community, while nodes in the second case are outliers. The nodes in the third case can be hubs, or the first node of an enclosing community group that is going to be merged one by one. However, at the time of merging a node, it is too early to judge whether the incoming node is a hub or not. Therefore, nodes in the first and third cases are merged into the community temporarily. After all qualified nodes are included, each node is re-examine by removing it from D and re-calculating the modularity L to only include the nodes in the first case. The remaining nodes are then constituent of the local community.

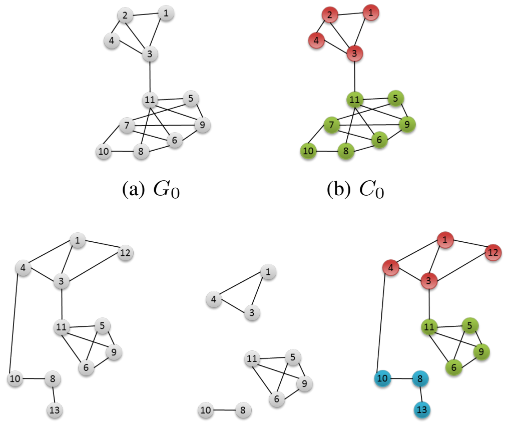
III. Incremental L-metric

Our proposed Incremental L-metric community mining considers the extracted connected components from communities of the previous snapshot as its initialization state. This is due to the fact that the activities and interactions of the entities frequently change and vary in time, the community found at snapshot $i-1$ may not result in a connected component in snapshot i. Thus, in order to use communities C_{i-1} in the process of detecting communities C_i, we first extract connected components from communities C_{i-1}. Then, the nodes at snapshot i are grouped based on the extracted connected components. And for each of the connected component $cc \in CC_i$, Algorithm 1 is executed iteratively. The connected components found are not only the members of the same communities at snapshot $i-1$, but also are connected to each other based on the interactions and connection at snapshot i. Each of these connected components are set as the seed for the L-metric community mining, where the algorithm construct its region D with the nodes of the given connected component. After that, the shell nodes of the region D have to be checked and if possible, added as the new community members. More specifically, a node v from the shell nodes is temporarily merged in the first and third cases into the community. After all qualified nodes are included, we re-examine each node by removing it from D and checking the metric value change if we merge it again. Now we only keep nodes if they are associated with the first case.

Algorithm 1 Incremental L-metric Community Mining

```
Input: $G_i$ (Graph at $i^{th}$ snapshot) and $cc$
Output: $D$ (cc's local community)
1. Discovery Phase:
  $D =$ all nodes from $cc$
  Add all boundary nodes of $D$ to $B$
  Add all external neighbours of $D$ to $S$
  repeat
    Find $v_m \in S$ with the maximum $L'$
    if $v_m$ belongs to the first or third case then
      add $v_m$ to $D$
    else remove $v_m$ from S
    end if
    Update B, S, C, L
  until $L' > L$
2. Examination Phase:
  for all $v \in D$ do
    Compute $L'$, keep $v$ only when it is the first case
  end for
```

A toy example to demonstrate the Incremental L-metric community mining is provided in Figure 2. At snapshot 0 (Figure 2a), L-metric detects the red and green communities (Figure 2b). To detect communities at snapshot 1 (Figure 2c), first we have to group the nodes based on the communities detected at snapshot 0. Finding the connected components, three groups of nodes are extracted (Figure 2d). Each of these three connected components are then the input of Algorithm 1, which results in the detection of the red, green, and blue communities at snapshot 1 (Figure 2e).



(a) G_0 (b) C_0

(c) G_1 (d) CC_1 (e) C_1

Fig. 2: Example: 2a and 2c show the structure of the network; 2d represents the connected components, while 2b and 2e illustrate the discovered communities.

IV. Experiment Results

In this section, we validate our proposed incremental community mining approach on the Enron email dataset. The Enron email dataset incorporates emails exchanged between employees of Enron Corporation. The entire dataset includes a period of 15 years and its corresponding email communication network, for the entire period of time, has over 80,000 nodes and several hundred thousand edges. We study the year 2001 and consider a total of 285 nodes and 23559 edges, with each month being one snapshot. For each of the 12 snapshots, one graph is constructed with the extracted employees as the nodes and email exchanged between them as the edges. Here, we compare the communities detected on Enron dataset by Incremental L-metric with the communities detected by Independent L-metric and FacetNet [12] algorithms. This comparison is performed from two perspectives: first, relatively based on a direct objective for dynamic communities, and then indirectly based on how much they improve the event detection framework.

A. Relative Evaluation

In the static scenario, the quality of a community mining result is mainly measured by Modularity Q [20]. However, in a dynamic scenario, the communities detected at one snapshot should not only be a good partitioning for that snapshot, but also a reasonable partitioning for the previous snapshot. Thus, we propose the Dynamic Modularity, DQ, to validate the quality of the partitioning on snapshot i defined as $DQ_i = \alpha Q(G_i, C_i) + (1-\alpha) Q(G_{i-1}, C_i)$. In this formula, $Q(G, C)$ computes the value of modularity Q for communities C based on the structure of graph G. Consequently, $Q(G_i, C_i)$ is computing the normal static Q modularity for communities discovered in snapshot i. While $Q(G_{i-1}, C_i)$ is the value of modularity Q for communities at snapshot i computed over graph from the previous snapshot. The average quality on all the snapshots, $\frac{1}{n}\sum_{i=1}^{n} DQ_i$, is then considered as the quality indicator for comparing different community mining algorithms. On the Enron dataset, we have the average score of 0.49, 0.44, 0.47 respectively for the Incremental L-metric, Independent L-metric and FacetNet, where $\alpha = .5$. Figure 3, presents a more detailed comparison of these three algorithms;

where quality, size, and number of communities over the time is depicted respectively. In top Figure, we can clearly see that the proposed incremental approach is consistently detecting communities with higher quality –complying with both current and temporal information. The other two Figures are shedding light on another difference between the incremental and independent approach. As we can see here, the average size of communities is much lower for the independent method. This is due to the fact that it failed to detect stable communities that span over time and instead detected several small communities, which is not surprising since it only looks at the current timeframe to mine communities. The FacetNet mining, fails similarly to the independent approach. One of the disadvantages of the FacetNet mining is that the number of communities should be similar for all the snapshots. As stated in [12], the number of communities is the one maximizing the average modularity over all the snapshots. For our results here, we run the FacetNet with different number of communities and chose 6, that resulted in the highest modularity.

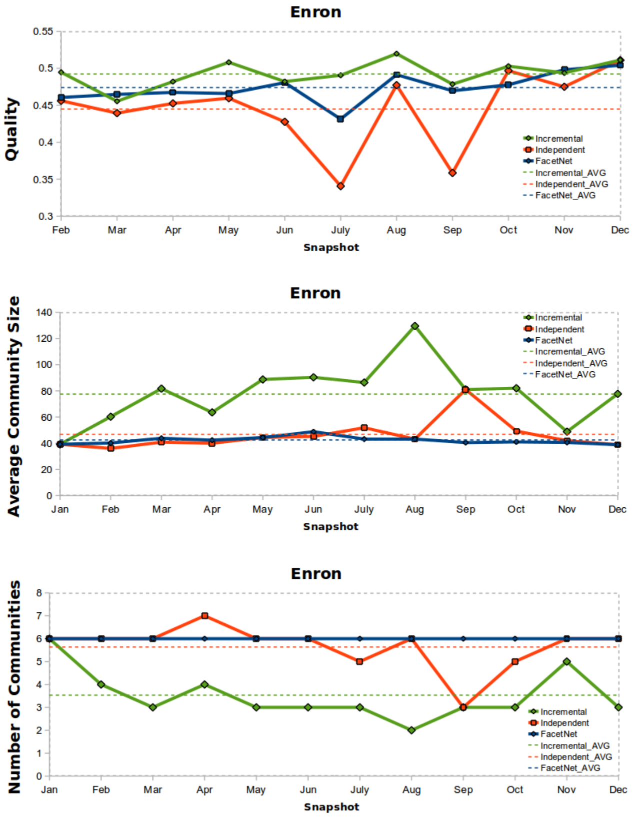


Fig. 3: Relative Evaluation: the dynamic modularity, size of communities, and the number of communities is reported for each snapshot, while the average over time for each method is represented by a constant dashed line.

B. *Indirect Evaluation*

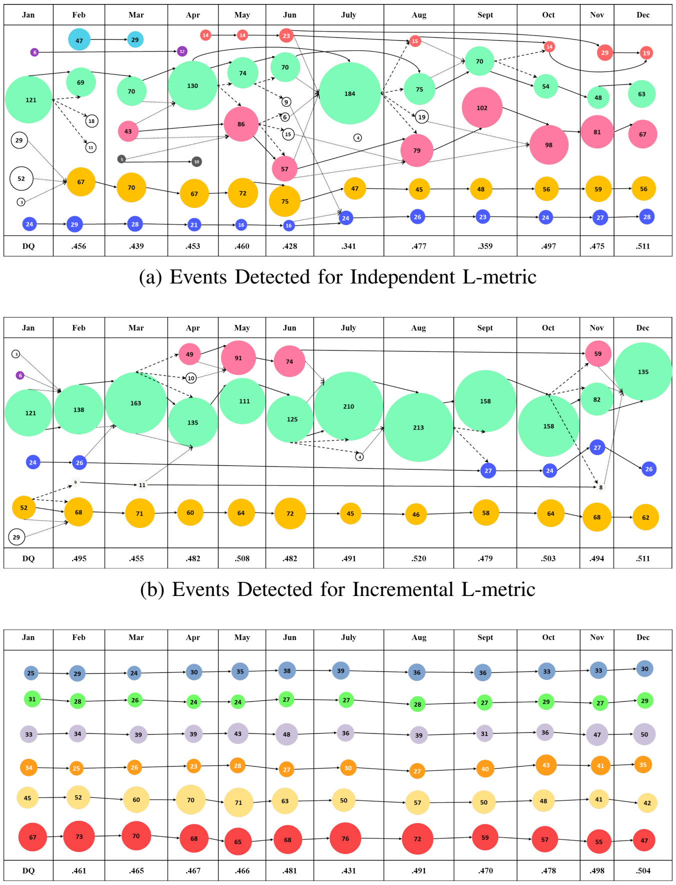
Here we compare the algorithms in the context of our event detection framework called MODEC [5], to see how accurate are the events detected based on their resulted communities. The comparison of events detected based on different community mining algorithms is shown in Table I, where the total number of events for each type detected during the 12 snapshots is provided. The Independent L-metric is too dynamic, detecting communities that vary much between snapshots, and therefore, resulting in too many triggered events, e.g. 19 forms, 19 dissolve. The FacetNet algorithm, on the other hand, is too stable, resulting in no merge or split events and only having survival events. Which is a consequence of how it detects communities over all the snapshots and has less emphasis on what is happening in each snapshot, and therefore fails to detect any of the events. The Incremental L-metric has *the balance* between the two, i.e. it correctly determines the communities survived over timeframes by incorporating the temporal information, and at the same time, detects other types of events reasonably.

TABLE I: Indirect Evaluation: comparison of Events Detected based on different algorithms (with the same parameter settings as [5]).

Framework	Form	Dissolve	Survive	Split	Merge	Mutual Topics
Incremental L-metric	10	10	32	5	8	4.12/10
Independent L-metric	19	19	46	7	11	3.83/10
FacetNet	6	6	66	0	0	4.02/10

The detailed communities and events detected are furthur shown in Figure 4. Here communities at each snapshot are marked with different colours, where these colours are the notion of meta communities. The communities detected by the Independent L-metric algorithm in Figure 4a, are too dynamic and unstable; which result in triggering too many events. For the first two snapshots for example, we can see that it failed to detect the green/largest community correctly, having that community as several separate smaller communities including the cyan/47 member community, which is not a distinct community and disappears after only one snapshot. The Incremental L-metric, Figure 4b, started with the same communities in the first snapshot, detects the survival of this green community correctly, by incorporating the temporal information. Its communities also have a relatively higher quality, with $DQ = .495$ to $DQ = .456$ of the independent method. The FacetNet communities are different than those found by the Independent and Incremental L-metric methods. And at the same time, have lower quality index of DQ. These communities are too stable and fail to trigger any events other than survival. Thus one is not able to see the patterns of change in the structure of the network using its detected communities.

In the Enron dataset, a community which *survives* multiple timeframes is more likely to continue discussions of the same topics. Therefore, we also incorporate the extraction of the topics for the discovered communities; where we apply KEA [21] to produce a list of 10 most frequent keywords discussed in the emails within 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 algorithm, which are reported at the last column of the Table I. Here, the highest mutual topics out of the top 10 most frequent keywords is obtained when using the Incremental L-metric framework. Thus, the Incremental L-metric also results in *the most meaningful* community evolution for Enron.



(a) Events Detected for Independent L-metric

(b) Events Detected for Incremental L-metric

(c) Events Detected for FaceNet

Fig. 4: Events Detected: solid, dashed, and dotted arrows show detected *survive*, *split*, and *merge* events respectively. Communities in (a)/(c) are too unstable/stable, while in (b) we have a balance between the change and stability. Also on average, the communities in (b) have a relatively higher quality compared to those in (a)/(c) –quality of communities detected in each snapshot is reported at the bottom of that snapshot.

V. Conclusion

One of the challenging research problems in dynamic social networks is to mine communities and analyze their evolution over the observation time. The traditional approach to solve this problem is to extract communities at each snapshot independent of the communities at other snapshots or the historic data. In this paper, we overviewed and classified different dynamic community mining approaches. We then proposed an Incremental L-metric community mining approach to consider both current and temporal data in the process of mining communities. The proposed method is then compared with its equivalent independent version and also with the most commonly used dynamic community method –FacetNet. Compared to these two methods, the Incremental L-metric method detects communities with higher quality when assessed directly with a modified version of Q modularity for the dynamic scenario. In addition, it is more successful in detecting the evolution patterns of the communities and triggering appropriate events, when used in our event detection framework, MODEC. The Independent L-metric is too unstable and triggers too many events, while the FacetNet is too stable and triggers no events other than survivals. Our incremental method, on the other hand, has the balance and provides meaningful communities and events by incorporating the temporal information.

References

[1] L. Backstrom, D. Huttenlocher, J. Kleinberg, and X. Lan, “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*, ser. KDD ’06, 2006.

[2] Edelman, “Edelman trust barometer report,” 2010.

[3] M. Takaffoli, F. Sangi, J. Fagnan, and O. R. Zaïane, “A framework for analyzing dynamic social networks,” in *Proceedings of the 7th Conference on Applications of Social Network Analysis*, ser. ASNA ’10, 2010.

[4] ——, “Modec - modeling and detecting evolutions of communities,” in *5th International AAAI Conference on Weblogs and Social Media*, ser. ICWSM ’11, 2011.

[5] ——, “Tracking changes in dynamic information networks,” in *Proceedings of the International Conference on Computational Aspects of Social Networks*, ser. CASoN ’11, 2011.

[6] J. Chen, O. R. Zaïane, and R. Goebel, “Detecting communities in large networks by iterative local expansion,” in *Proceedings of the International Conference on Computational Aspects of Social Networks*, ser. CASoN ’09, 2009.

[7] G. Palla, A.-L. Barabási, and T. Vicsek, “Quantifying social group evolution,” *Nature*, vol. 446, no. 7136, pp. 664–667, 2007.

[8] S. Asur, S. Parthasarathy, and D. Ucar, “An event-based framework for characterizing the evolutionary behavior of interaction graphs,” *ACM Transactions on Knowledge Discovery from Data*, 2009.

[9] Y. Wang, B. Wu, and N. Du, “Community evolution of social network: Feature, algorithm and model,” *Science And Technology*, 2008.

[10] D. Greene, D. Doyle, and P. Cunningham, “Tracking the evolution of communities in dynamic social networks,” in *Proceeding of International Conference on Advances in Social Networks Analysis and Mining*, ser. ASONAM ’10, 2010.

[11] D. Chakrabarti, R. Kumar, and A. Tomkins, “Evolutionary clustering,” in *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, ser. KDD ’06, 2006.

[12] Y.-R. Lin, Y. Chi, S. Zhu, H. Sundaram, and B. L. Tseng, “Analyzing communities and their evolutions in dynamic social networks,” *ACM Transactions on Knowledge Discovery from Data*, 2009.

[13] S. Yu, K. Yu, and V. Tresp, “Soft clustering on graphs,” in *The Neural Information Processing Systems*, ser. NIPS, 2005.

[14] T. B.-W. Chayant Tantipathananandh, “Finding communities in dynamic social networks,” in *Proceedings of 11th IEEE International Conference on Data Mining*, ser. ICDM ’11, 2011.

[15] Y. Sun, J. Tang, J. Han, M. Gupta, and B. Zhao, “Community evolution detection in dynamic heterogeneous information networks,” in *Proceedings of the Eighth Workshop on Mining and Learning with Graphs*, ser. MLG ’10, 2010.

[16] C. C. Aggarwal and P. S. Yu, “Online analysis of community evolution in data streams,” in *Proceedings of SIAM International Data Mining Conference*, ser. SDM’05, 2005.

[17] M.-S. Kim and J. Han, “A particle-and-density based evolutionary clustering method for dynamic networks,” *Proceedings of the VLDB Endowment*, August 2009.

[18] T. Aynaud and J.-L. Guillaume, “Static community detection algorithms for evolving networks,” in *WiOpt Workshop on Dynamic Networks*, 2010.

[19] V. D. Blondel, J.-L. Guillaume, R. Lambiotte, and E. Lefebvre, “Fast unfolding of communities in large networks,” *Journal of Statistical Mechanics: Theory and Experiment*, 2008.

[20] M. E. J. Newman, “Modularity and community structure in networks,” *Proceedings of the National Academy of Sciences*, 2006.

[21] I. H. Witten, G. W. Paynter, E. Frank, C. Gutwin, and C. G. Nevill-Manning, “KEA: Practical automatic keyphrase extraction,” in *ACM DL*, 1999.