Analysis on Community Variational Trend in Dynamic Networks

Xiaowei Jia, Nan Du, Jing Gao, Aidong Zhang
State University of New York at Buffalo, Buffalo, NY, USA
{xiaoweij,nandu,jing,azhang}@buffalo.edu

ABSTRACT

Temporal analysis on dynamic networks has become a popularly discussed topic today, with more and more emerging data over time. In this paper we investigate the problem of detecting and tracking the variational communities within a given time period. We first define a metric to measure the strength of a community, called the normalized temporal community strength. And then, we propose our analysis framework. The community may evolve over time, either split to multiple communities or merge with others. We address the problem of evolutionary clustering with requirement on temporal smoothness and propose a revised soft clustering method based on non-negative matrix factorization. Then we use a clustering matching method to find the soft correspondence between different community distribution structures. This matching establishes the connection between consecutive snapshots. To estimate the variational rate and meanwhile address the smoothness during continuous evolution, we propose an objective function that combines the conformity of current variation and historical variational trend. In addition, we integrate the weights to the objective function to identify the temporal outliers. An iterative coordinate descent method is proposed to solve the optimization framework. We extensively evaluate our method with a synthetic dataset and several real datasets. The experimental results demonstrate the effectiveness of our method, which is greatly superior to the baselines on detection of the communities with significant variation over time.

Categories and Subject Descriptors

H.2.0 [**Information Systems**]: General

Keywords

Dynamic Networks; Evolutionary Clustering; Community Analysis

CIKM'14, November 3–7, 2014, Shanghai, China.

http://dx.doi.org/10.1145/2661829.2662004 .

1. INTRODUCTION

Dynamic network analysis aims at discovering knowledge from evolutionary process of network data. It has attracted more and more attention in recent years, with the availability of many dynamic online datasets. Compared with static network analysis focusing on only one snapshot, dynamic analysis makes it possible to detect the key elements through a certain period of time, or the critical factors to the evolutionary trend [25]. It also has extensive prospect in many real applications [4, 23], such as future prediction, investment and so on.

In this paper we focus on dynamic community variation analysis. There have been many studies in recent years on community level research [1,6,11,17,24,26]. Compared with node level analysis, the community level work widens our scope in the exploration of group formation and structure. We propose a novel framework to detect the communities that have significant strength variation. We first define the normalized community strength by claiming that, the strength of a network should depend on the difference between the internal and outward edge weights. And then, we take into account the strength of historical networks to guarantee that the network progresses smoothly.

To make the clustering well suited for dynamic evolution scenario, we adopt a revised method based on non-negative matrix factorization(NMF) [12]. In many real world networks, one node may simultaneously belong to multiple communities, e.g., an author working in an interdisciplinary research could have frequent collaboration with people in different areas. For this reason, we propose our community detection method based on NMF which allows overlapping clusters. Besides, we combine the current clustering cost with the historical cost to jointly measure the clustering quality in a dynamic process. After the communities are detected from the network at time t, we will use them as the initialization for time $t+1$. This guarantees the clustering between consecutive timestamps will not change dramatically. This can also be well explained in an evolutionary view, where the clustering is formed as a result of the previous snapshot, and evolves into new community structure by an iterative update process. After we obtain the clusterings on each snapshot, we find the correspondence between each consecutive snapshots by solving an optimization function. With the knowledge of corresponding matching, the individually detected communities from each snapshot can be linked as a whole series. By the above processing, we can deal with the evolving communities that allow membership change in further investigation and analysis.

After we obtain all the clustering and correspondence information, we are able to detect the communities with significant variations. In this paper, we focus on the variation in community strength. To illustrate this kind of variation, a toy example of a user-to-user network is given in Figure 1. In this example the network consists of two communities, with nodes from each one represented in either circle or diamond shapes. At time t in Figure 1(a), both of communities have exactly the same strength. Here we show three principal kinds of evolutionary changes in strength, by different conditions at time $t+1$. For case 1 at time $t+1$ shown in Figure 1(b), the users in the left community have more interactions than previous timestamp while the users in the right community lose some internal links that existed before. Hence we can determine the relatively high variational rate of the left community. For case 2 shown in Figure 1(c), one user from the right community joins the left community, which changes the memberships in both communities. This case can be further extended to community splitting and merging. For case 3 shown in Figure 1(d), the users in both communities have more interactions than before. However, with comparison we can observe that the left community grows much faster than the right one. In real applications there may exist a global trend that leads to an overall unified variation. In this case we are more interested in communities with significant growth or recession compared with this global trend.

From the detected communities we can discover lots of interesting knowledge. This can be well illustrated by many real life scenarios. It is easy to notice in social networks that some newly emerging groups grow rapidly, with members becoming more interactive as time progresses. This is probably caused by the fact that these members share common interests, which are quite related with the current period, such as election or the advent of some new social phenomena. In social-sharing networks, a fast growing community usually indicates the viral spread of certain messages. It is very likely to discover hot research topics in a co-author network if we can detect the community with rising collaboration frequency. We can also find similar cases in the biological domain. For certain kind of diseases such as cancer, it has been reported [10] that the expression profiling of some genes will change as the cancer progresses. If we can find the communities with significant changes compared with others, it will be a remarkable insight and improvement to the medical research.

The variational rate should depend on the strength change from one time to the next. In addition, we consider temporal smoothness of the variational rate by combining with historical growth trend, as we do not expect it to change dramatically. Furthermore, there are often noise caused by either uncommon perturbation or data acquisition error. To reduce the influence of these factors, we apply weights on communities to estimate the contribution made by each of them. Given these conditions, we propose an objective function, which integrates temporal information and noise reduction. By solving this function using an iterative updating method, we can figure out the final strength variational rate.

We organize the paper as follows. In Section 3.1, we define the community strength metric. In Section 3.2, we propose the first stage of the framework, and describe our method on solving evolutionary clustering and soft clustering correspondence. Section 3.3 presents the second stage with the measurement and detection of community variation. Some extensive applications are listed in Section 3.4. We include our experimental results in Section 4 and we conclude our work in Section 5.

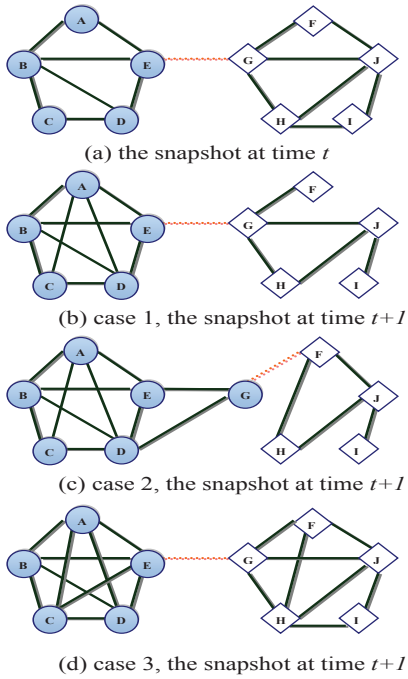


Figure 1: A Toy Example of Variation in Community Strength.

2. RELATED WORK

The problem of community detection in dynamic networks has been studied in recent years [3, 5, 7, 8, 18, 22, 27]. Compared with static networks, the clustering on dynamic networks should consider historical conformity and temporal factors. *Chi et al.* [7,8], *Lin et al.* [18], *Chakrabarti et al.* [5] and *Berger-Wolf et al.* [3] proposed different community discovery methods on dynamic networks. These methods mainly combined the clustering quality on both current and historical networks. Even if these methods can guarantee the accuracy and consistency for clustering results, they provide limited information for each community during the evolutionary process.

There are also many works devoted in solving the correspondence between different clusterings. The clustering describes the community layout in each snapshot. The matching can be implemented in a hard way [15], where only the best matched entity is selected. This can be applied to the problem of label switching, but not with the dynamic network analysis, which involves community merging and splitting. To allow one-to-many relationships, the clustering matching can be done in a soft way [20], in which each entry of the matching matrix denotes the probability of the correspondence between the pair of communities. *Gupta et al.* [14] also proposed a method to do community matching with the awareness of outliers. It reduces the affect brought by the temporal outliers on the matching problem by solving an objective function in form of weighted square error.

With recent research on dynamic networks, we can have a better idea on how users behave in networks and how communities evolve over time. *Backstrom et al.* [2] did research on the evolution and growth of communities in social networks. They analyzed the features that lead to the trend of community growth. *Cook et al.* [9] reported that group chat on Twitter leads to a growing trend. However, these works do not well address the issue of how to precisely detect and track the communities that vary significantly as time progresses. *Nan et al.* [13] proposed the PACS algorithm, which measures the temporal community strength by combining both current and historical network information. Even if PACS algorithm provides an effective way to measure the community strength, it can only apply to the case with fixed community memberships. If we track in a long period with many membership alterations involved, the result from PACS will be less meaningful. Further, it can not be used to measure community variation.

3. COMMUNITY VARIATION DETECTION APPROACH

The basic flow of our framework is given in Figure 2. It can be divided into two stages. The first stage involves solving evolutionary clustering in dynamic networks and correspondence between snapshots. Then by combining the strength in each community and the matching information, we use the proposed algorithm to estimate the variational rate for each community. In this section, we will first define the

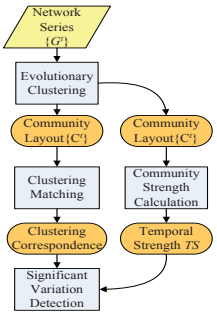


Figure 2: The Flow of Whole Framework.

temporal community strength, based on which we will propose our dynamic community significant variation detection approach. To formalize the problem, we first start with the notation introduction. We represent a matrix using the notation D. $D_{i,j}$ denotes the $(i,j)^{th}$ element of matrix D. $D_{i.}$ and $D_{.j}$ denote the i^{th} row and j^{th} column of matrix D, respectively. As for a vector V, V_i represents the i^{th} element in this vector. Some notations used in this paper are listed in Table 1.

3.1 Defining Temporal Community Strength

Given a network sequence $\{G^t\}_{t=1,..,T}$, each $G^t =< V^t, E^t, W^t >$ is the network at time t, where V^t is the set of nodes in the network, E^t includes all the edges that link pair of nodes, and W^t is a symmetric matrix representing the edge weights. Each community in network at time t is a subset of V^t, which usually has strong internal intensity compared with between-community connections. The community strength

Table 1: Table of Notation

Notation	Meaning
$W^t_{N\times N}$	weight matrix at time t
$C^t_{N\times K_t}$	membership matrix at time t
$M_{K_{t_1}\times K_{t_2}}$	clustering matching matrix from t_1 to t_2
$TS_{K\times T}$	$TS_{z,t}$: temporal strength of community z at t
$R^t_{K^t\times 1}$	variational rate of communities at time t
$t = 1, ..., T$	index for snapshot
$k = 1, ..., K_t$	index for community at time t

of a community z in G_t can be defined as [13]:

$$Strength(z) = \sum_{i\in V^t}\sum_{j\in V^t} w_{ij} * \sum_{k\in z}\sum_{l\in z} w_{kl} - (\sum_{k\in z}\sum_{m\in V^t} w_{km})^2, \tag{1}$$

by which we can conclude that the communities with higher strength should have stronger internal intensity and less outward edges. The first term $\sum_{i\in V_t}\sum_{j\in V_t} w_{ij}$ is used to ensure the positive strength value.

As usually well detected communities have much stronger internal intensity, the penalty from the external edges, or the second term in Eq. 1 is limited. The results are thus biased towards the ones with larger size. Here we propose the normalized community strength as:

$$NorStrength(z) = \frac{Strength(z)}{|z|}, \tag{2}$$

which penalizes the strength value by the size of community. It can be further reformulated with some matrix derivation to the following equation:

$$NorStrength(z) = \frac{{C^t_{.z}}^T(\sum(W^t) * W^t - D)C^t_{.z}}{|z|} \quad \text{where} \quad D = W^t * \mathbf{1}_{|V^t|,|V^t|} * {W^t}^T, \tag{3}$$

in which C^t is the $|V^t|$–by–K_t cluster membership matrix at time t, with $(i,j)^{th}$ element denotes the probability that the i^{th} node belongs to j^{th} community. K_t denotes the number of clusters detected at time t. $\mathbf{1}_{N_1,N_2}$ denotes an N_1–by–N_2 matrix with all entries set as 1.

In real applications, people usually expect gradual transition of community strength, which necessitates a tradeoff between current and historical status. The strength value can then be reformulated by solving the following objective function:

$$\begin{aligned}\min_{TS_{.t}} J(TS_{.t}) = \phi \sum_z log(\frac{1}{TS_{z,t}}) NorStrength(z) \\ +(1-\phi)\sum_z log(\frac{1}{TS_{z,t}}) NorHStrength(z) \\ s.t. \sum TS_{.t} \le \mu_t, TS_{z,t} \ge 0, \forall z = 1, ..., K_t,\end{aligned} \tag{4}$$

in which TS is the matrix of temporal community strength, with $(z,t)^{th}$ element representing the temporal community strength of z at time t. By using Lagrangian Multipliers, we can rewrite the objective function as:

$$\begin{aligned}\min_{TS_{.t}} J(TS_{.t}) = \phi \sum_z log(\frac{1}{TS_{z,t}}) NorStrength(z) \\ + (1-\phi) \sum_z log(\frac{1}{TS_{z,t}}) NorHStrength(z) \\ + \gamma(\sum_z TS_{z,t} - \mu_t).\end{aligned} \tag{5}$$

Here we only focus our optimization with respect to a certain timestamp t. $NorStrength(z)$ is the normalized community strength of z, and $NorHStrength(z)$ is the normalized community strength of z on previous snapshot, that is:

$$NorHStrength(z) = \frac{{C^t_{.z}}^T(\sum(W^{t-1}) * W^{t-1} - D^{t-1})C^t_{.z}}{|z|} \quad \text{where} \quad D^{t-1} = W^{t-1} * \mathbf{1}_{|V^{t-1}|,|V^{t-1}|} * {W^{t-1}}^T, \tag{6}$$

By setting the derivative of the equation with respect to $TS_{z,t}$, we obtain the following equations:

$$TS_{z,t} = \frac{\phi NorStrength(z) + (1-\phi)NorHStrength(z)}{\gamma} \tag{7}$$

and

$$\gamma = \frac{\sum_z[\phi NorStrength(z) + (1-\phi)NorHStrength(z)]}{\mu_t} \tag{8}$$

Then from equations above we get the solution for $TS_{z,t}$ as:

$$TS_{z,t} = \frac{[\phi NorStrength(z)+(1-\phi)NorHStrength(z)]\mu_t}{\sum_z[\phi NorStrength(z)+(1-\phi)NorHStrength(z)]} \tag{9}$$

3.2 Evolutional Clustering and Matching

For each network at time t, i.e., G^t, we first conduct community detection on the network, and then match the consecutive clusterings with soft correspondence. This procedure can guarantee the accuracy of clustering and meanwhile establish the links among communities from different timestamps.

To allow for soft clustering, we adopt the 3-factor NMF. To make sure the clustering does not change dramatically from time $t-1$ to time t, we penalizes the clustering quality of current clustering on the previous network. As the input network matrix is symmetric in our case, the optimization function is as follows:

$$\min_{C^t\geq 0, S^t\geq 0, S^{t\prime}\geq 0} \|W^t - C^t S^t {C^t}^T\|^2 + \alpha\|W^{t-1} - C^t S^{t\prime} {C^t}^T\|^2, \quad s.t. {C^t}^T C^t = I, \tag{10}$$

where α denotes the weight for penalty of current clustering on historical network, C^t is the $|V^t|-by-K_t$ cluster membership matrix, K_t is the number of clusters at t, and S^t is the cluster relationship matrix. The iterative update rules for C^t, S^t and $S^{t\prime}$ are as:

$$c^t_{ij} = c^t_{ij}\sqrt{\frac{(W^{t^T}C^tS^{t^T} + \alpha W^{t-1^T}C^tS^{t\prime T})_{ij}}{(C^tC^{t^T}(W^{t^T}C^tS^{t^T} + \alpha W^{t-1^T}C^tS^{t\prime T}))_{ij}}} \tag{11}$$

$$s^t_{ij} = s^t_{ij}\sqrt{\frac{(C^{t^T}W^{t^T}C^t)_{ij}}{(C^{t^T}C^tS^tC^{t^T}C^t)_{ij}}} \tag{12}$$

$${s^t_{ij}}' = {s^t_{ij}}'\sqrt{\frac{(C^{t^T}W^{t-1^T}C^t)_{ij}}{(C^{t^T}C^tS^{t\prime}C^{t^T}C^t)_{ij}}}. \tag{13}$$

We update one parameter by fixing the others and repeat the process iteratively until convergence. Given the condition in real applications, we can assume that the communities evolve continuously, i.e., the formation of a group should depend more or less on historical data. We do not consider the situation in which events take place independently at each time or with random probability. Based on this assumption, after we compute the community membership matrix for a certain timestamp t, we set it as the initialization for time $t+1$, which ensures the smoothness of clustering evolution. The whole procedure is given in Algorithm 1.

Algorithm 1: Evolutional Clustering

Input: A series of network $\{G^t\}_{t=1,..,T}$
Output: community membership $\{C^t\}_{t=1,..,T}$

```
Initialize C^0, S^0 and S^0';
for t ← 0 to T − 1 do
    while not convergence do
        Update S^t by Eq. 12;
        Update S^t' by Eq. 13;
        Update C^t by Eq. 11;
    {C^{t+1}, S^{t+1}, S^{t+1'}} = {C^t, S^t, S^t'};
return {C^t}_{t=1,..,T};
```

The result of Algorithm 1 is a series of community membership matrices $\{C^t\}_{t=1,..,T}$. Even if we have initialized the clustering process by previous clustering, the clusterings in different timestamps may still be inconsistent with their cluster labels, e.g.,C^t and C^{t+1} are as follows:

$$C^t = \begin{pmatrix}1 & 0\\ 1 & 0\\ 0 & 1\end{pmatrix} \qquad C^{t+1} = \begin{pmatrix}0 & 1\\ 0 & 1\\ 1 & 0\end{pmatrix}.$$

It is then very likely that the first cluster in C^t corresponds to the second cluster in C^{t+1} and vice versa. Also, it is commonly observed that one community may split into multiple while several other communities can merge into one during evolution. Thus, a clustering matching is necessary for consistency maintenance. Given the assumption that networks evolve gradually and smoothly, we can focus on the clustering matching between two consecutive snapshots.

Given two clusterings represented by membership matrices as C^{src} and C^{tar}, we are going to find the matching relationship M from C^{src} to C^{tar} by solving the following optimization function, which is proposed in [20]:

$$\begin{aligned} f(M) = &\|C^{tar} - C^{src}M\|^2 \\ &- \xi\|M - \frac{1}{K_{src}}\mathbf{1}_{K_{src}K_{src}}M\|^2 \\ &+ \beta\|M\mathbf{1}_{K_{tar}K_{tar}} - \mathbf{1}_{K_{src}K_{tar}}\|^2 \\ &s.t. M_{i,j} \geq 0, \forall i = 1,...,K_{src}, \forall j = 1,...,K_{tar}. \end{aligned} \tag{14}$$

The second term in the function controls the sparseness of matching, which very well fits our case of dynamic evolution. As in consecutive timestamps merging and splitting usually occurs gradually and gently, we use the sum of variance on the matching probability for each cluster in C^{tar} to balance the correspondence, which guarantees that only limited number of communities from C^{src} are matched to each community in C^{tar}. ξ denotes the weight for this term. The third term serves as the approximation of the constraint that $\sum_j M_{i,j} = 1$ and β denotes the weight for this term. M_{z_1,z_2} represents the probability that cluster z_1 in C^{src} matches to cluster z_2 in C^{tar}.

We use the following multiplicative updating rule:

$$M = M \circ \frac{C^{srcT}C^{tar} + \beta K_{tar}\mathbf{1}_{K_{src}K_{tar}}}{E} \tag{15}$$

$$E=C^{src^T}C^{src}M-\xi M+\frac{\xi}{K_{src}}\mathbf{1}_{K_{src}K_{src}}M+\beta K_{tar}M\mathbf{1}_{K_{tar}K_{tar}} \tag{16}$$

The symbol of '$\circ$' denotes the entry-wise matrix product. It has been proved by utilizing auxiliary function [20] that the value of $f(M)$ in Eq. 14 will be non-increasing and will finally reach convergence by the update using Eq. 15. For each pair of consecutive community membership matrix C^t and C^{t+1}, we iteratively update M until convergence. With the community matching matrix, we can have a clear idea how the community of one time matches to the other in the next timestamp, and even how communities merge or split to form new ones during evolution.

3.3 Community Strength Variation Analysis

We are interested in the communities with significant variation in their community strength. On one hand, we analyze in a dynamic view to detect this type of phenomena, as this occurs during the evolutionary process. On the other hand, the growth or recession should not depend too much on the size of community or the strength of community. We can illustrate this problem with a social network example. There are many topics that have been liked or shared by a large number of people. However, these groups may stay with relatively static activities over time and show little evidence of further growth. On the contrary, there may be some small groups where apparent growth takes place with their interactions. We are more interested in these groups, where more useful knowledge can be discovered, and some measures can be taken accordingly based on predictable future trend.

We investigate the problem of community growth by considering the relationship between consecutive timestamps. In a real evolutionary process, we hold the assumption that the community strength value varies gradually and the probability of sharp increase or decrease is rare. Hence we measure the variational rate as a linear combination of the change at current time with the previous trend. On the other hand, it is commonly seen that some perturbation takes place with data over time in real applications. This kind of noise may be detected by traditional methods as significant growing or recession phenomena and mislead the learning method. To mitigate these effects, we use weight parameters to control the contribution for each community, and propose the following objective function with respect to a certain timestamp t:

$$\begin{aligned}F_{a_k,R}=&\sum_k log(\frac{1}{a_k})[\theta(TS_{k,t}-\sum_{k'\in L^1}M_1(k,k')TS_{k',t-1}R_{k'})^2\\&+(1-\theta)(\sum_{k'\in L^1}M_1(k,k')TS_{k',t-1}-\sum_{k'\in L^1}M_1(k,k')\\&*(\sum_{k''\in L^2}M_2(k',k'')TS_{k'',t-2})R_{k'})^2],\\&s.t.\sum_k a_k\le\mu_1,\sum_k R_k=\mu_2,R_k\ge 0,0\le a_k\le 1,\\&\forall k=1,...,K_t\end{aligned} \tag{17}$$

in which $log(\frac{1}{a_k})$ is the weight to control the contribution for each community, θ is the parameter for the linear trade-off between current and previous variation, M_1 stands for the matching matrix of clustering from time t to $t-1$, M_2 stands for the matching matrix from time $t-1$ to $t-2$, and here we use $M(i,j)$ to denote the $(i,j)^{th}$ entry in M. L^1 and L^2 denote the set of communities at time $t-1$ and $t-2$, respectively, and R_z represents the variational rate of community z at t, the time when z emerges. Then we solve the optimization problem by following the iterative method.

We first use Lagrangian Multipliers to rewrite the objective function together with the constraint, φ is the Lagrangian multipliers for constraint $\sum_k a_k\le\mu_1$ and λ is the one for constraint $\sum_k R_k=\mu_2$. Then by setting the derivative to zero with respect to a_k, we have the following updating rule:

$$a_k=\frac{GD(k)}{\varphi}, \tag{18}$$

$$\begin{aligned}\text{where}\quad GD(k)=&\theta[TS_{k,t}-\sum_{k'\in L^1}M_1(k,k')TS_{k',t-1}R_{k'}]^2\\&+(1-\theta)[\sum_{k'\in L^1}M_1(k,k')TS_{k',t-1}-\sum_{k'\in L^1}M_1(k,k')\\&*(\sum_{k''\in L^2}M_2(k',k'')TS_{K'',t-2})R_{k'}]^2,\end{aligned} \tag{19}$$

and

$$\varphi=\frac{\sum_k(GD(k))}{\mu_1}. \tag{20}$$

By combining Eq. 18 and Eq. 20, we have:

$$a_k=\frac{\mu_1 GD(k)}{\sum_k GD(k)}. \tag{21}$$

Then we update R from a_k. By setting the derivative with respect to R_z as zero, we have the following updating rules:

$$R_z=\frac{NN(z)}{DN(z)}, \tag{22}$$

$$\text{where}\quad DN(z)=\sum_k log(\frac{1}{a_k})[2\theta TC(z)^2+2(1-\theta)TP(z)^2], \tag{23}$$

$$\begin{aligned}TC(z)&=M_1(k,z)TS_{z,t-1}\\TP(z)&=M_1(k,z)\sum_{k'}M_2(z,k')TS_{k',t-2},\end{aligned} \tag{24}$$

$$\text{and}\quad NN(z)=\sum_k log(\frac{1}{a_k})[2\theta TCC(z)+2(1-\theta)TPP(z)]-\lambda, \tag{25}$$

$$\begin{aligned}TCC(z)&=[TS_{k,t}-\sum_{k'\in L^1\setminus z}M_1(k,k')TS_{k',t-1}R_{k'}]*TC(z)\\TPP(z)&=[\sum_{k'\in L^1}M_1(k,k')TS_{k',t-1}-\sum_{k'\in L^1\setminus z}M_1(k,k')\\&\quad*(\sum_{k''\in L^2}M_2(k',k'')TS_{k'',t-2})R_{k'}]*TP(z),\end{aligned} \tag{26}$$

To formulate the method in matrix operations, we have listed the whole process in Algorithm 2 with some derivation and name it as **VACS**(Variation Analysis of Community Strength). In the algorithm, TC, TP, DC, DP, TCC and TPP are used as some intermediate matrices or vectors to reduce repetitive computing during the calculation of DN and NN, which are derived from Eq. 23 and Eq. 25. The symbol of '$\oslash$' denotes entry-wise division. In general, the variational rate will be larger than 1 for the growth in community strength, and less than 1 for the recession.

Our proposed method **VACS** well utilizes the information from evolutionary clustering and soft matching correspondence. It allows changes in community membership and takes account of temporal smoothness. Also by the weight setting and the bound on $\sum R$, large portion of the influence

Algorithm 2: VACS: Community Growth Analysis

Input: Community Strength TS, Clustering Matching Matrices
Output: a, R

Initialize R;
while *not converge* **do**
  Calculate TC and TP by Eq.24:
  TC=$TS_{.t}$− M_1*($TS_{.t-1} \circ R$);
  $TP = M_1 * TS_{t-1.} - M_1 * ((M_2 * TS_{t-2.}) \circ R)$;
  Calculate a by Eq.21:
  $S = \theta * TC \circ TC + (1-\theta) * TP \circ TP$;
  $a = \mu_1 S \oslash sum(S)$;
  Calculate DN by Eq.23:
  $LGA = log(1 \oslash a)$; $DC = M_1 * diag(TS_{.t-1})$;
  $DP = M_1 * diag(M_2 * TS_{.t-2})$;
  $DN = LGA^T * (2\theta * DC_{.i} \circ DC_{.i} + 2(1-\theta) DP_{.i} \circ DP_{.i})$;
  for *each community* i **do**
    Calculate NN by Eq.25 and Eq.26:
    $TCC = diag(TC) * DC + DC * diag(R) \circ DC$;
    $TPP = diag(TP) * DP + DP * diag(R) \circ DP$;
    Calculate λ using Eq.22 and constraints in Eq.17;
    $NN = LGA^T * (2\theta TCC_{.i} + 2(1-\theta) TPP_{.i}) - \lambda$;
    Calculate $R_i = NN/DN$;
return a, R;

from temporal outliers can be mitigated. After we solve the objective function at time t, we can obtain the variational rate R_z for all z belonging to snapshot at $t-1$. The design at this point is reasonable due to the fact that, by the time t, we only have enough information to estimate the growing trend for communities at $t-1$, with the change during the interval from $t-1$ to t. On the other hand, for optimization procedure on each R_z, we need the matching and temporal strength information from all communities at time t. This explains the reason why we segment our framework as two stages and conduct clustering and matching before the estimation of variational rate on communities.

3.4 Extensive Applications

The proposed framework can estimate the variational rate for each community detected in dynamic networks. In addition, the method can be further extended to many applications. Here we introduce the problem of top-K consistently growing communities detection and online incremental processing based on our method.

Top-K most Consistently Growing Communities.

Through Algorithm 2, we have estimated the variational rate for each community. In most cases, compared with the communities that suddenly shoot up, people are more interested in the ones with consistent growth over whole tracking period. As the variational rate is highly related with the ratio of strength between consecutive snapshots, we aggregate them by the product of the variational value at each snapshot. Assume the community z appears at time t_z, the growth consistency value can thus be expressed as:

$$GC_z = ...RM_{t_z-2} * RM_{t_z-1} * R_z * RM_{t_z+1} * RM_{t_z+2}.....$$
$$\text{where} \quad RM_{t_z-n} = M_{1z.} M_2 ... M_n R^{t_z-n}, \tag{27}$$

in which R^t is a column vector which consists of the variational value of all the communities detected at time t, M_n denotes the clustering matching matrix from time $t-n+1$ to time $t-n$ when n is positive, and the one from time $t-n-1$ to $t-n$ when n is negative.

We can also formulate Eq. 27 in a weighted way, and the measure becomes $GCM_z = \sum_t w_t * log(RM(t))$, where $RM(t_z) = R_z$. The weight w_t can depend on many factors, such as the interval length from timestamp t to $t+1$, the number of communities at time t, the overall average growth value at time t, and so on.

Online Community Growth Analysis.

In recent years, online analysis has attracted more and more attention. Online processing means we should run our algorithm on streaming data without future information. In this part we briefly discuss the online version of **VACS**. It is very helpful if we can detect the fastest or slowest growing community at the time when it happens, by which we can take some measures accordingly as soon as possible.

The **VACS** method can be easily adapted in online incremental applications. Assume we are working on a parallel computing environment. When new data arrives at time t, we already have all the information before t. We can calculate evolutionary clustering from C^{t-1} by solving Eq. 10. The NMF problem can be efficiently solved in parallel [19]. Then we compute in parallel on each community in C^t for its matching to C^{t-1} as well as its temporal community strength. After synchronization, we solve Eq. 17 iteratively on each community emerging at time $t-1$ in parallel. An outer non-parallel loop is used to control the update for $\{a_k\}$. The parallel update for each R^i is implemented inside the loop. When we reach convergence, we calculate the estimate of variational rate for each community at time t as $Re_i = M_{i.} * R^{t-1}$, where M is the matching matrix and R^{t-1} is the variational rate for communities arising at time $t-1$, i.e., the output from **VACS**. Then we use Re as the initialization for the iterative calculation for R^t, when new data comes.

4. EXPERIMENTS

In this section we implement our framework on both synthetic and real world datasets. We validate the performance of community detection in first two experiments, by comparing with ground truth information. In the second and third experiments we evaluate our method on DBLP and IMDB datasets as case studies. We justify our results by using concrete data information.

4.1 Synthetic Dataset

In this part we first evaluate our method on a synthetic dataset created according to [18]. We generated networks for 10 consecutive snapshots, each with 100 nodes. We manually divide them into 5 communities $\{C_1, C_2, C_3, C_4, C_5\}$ and each has 20 nodes. At the initial stage, we add edges for each community by tuples (p_{in}, p_{out}) and set the weight on each edge to be 1. p_{in} denotes the probability to add each internal edge and p_{out} stands for the probability to add each between-community edge. We set the value for each community as $(0.2, 0.1)$ and generate the network in 1^{st} snapshot. Then from 2^{nd} to 10^{th} snapshot, we increase or decrease the weight on edges by using a 4-tuple $G = (p_{inc}, p_{dec}, p_{oinc}, p_{odec})$. From snapshot $t-1$ to t, we double the weight on each internal edge with probability p_{inc}, or add the edge if the weight is 0 at $t-1$. On the contrary, for each internal edge, we reduce the weight by half with probability p_{dec}. Similarly, we use p_{oinc} and p_{odec} to control weight increase or decrease for between-community edges. We set this 4-tuple to each community as $C_1 = (0.47, 0.1, 0.1, 0.3)$, $C_2 = (0.35, 0.17, 0.12, 0.2)$,

$C_3 = (0.32, 0.18, 0.1, 0.1)$, $C_4 = (0.32, 0.19, 0.1, 0.13)$, $C_5 = (0.27, 0.23, 0.1, 0.05)$. As this dynamic network is manually created, we know the community C_1 has the most significant growing trend and C_5 has the slowest variation. To show the effectiveness of our method, we compare our method with two baseline methods. As there is no previous method that handles the same problem, we make comparison with the variation of two previous relevant methods [13].

Community Internal Density(CID): In this baseline method, we first calculate the internal density for each community by the normalized sum of its internal weight [17], as $CID_z^t = \frac{\sum_{i,j \in z} w_{ij}}{|z|(|z|-1)/2}$. We calculate the rate of change as $Rate_C(z) = CID_{t+1}(z)/CID_t(z)$.

K-Nearest Neighbors(KNN): At each snapshot, for each community z we detect its top-K most similar community by Jaccard coefficient. Then the variation can be computed as the weighted sum of the variation rate of these K communities, $Rate_K(z) = \sum_i h_{zi} R_i$. The weight h_{zi} is the normalized Jaccard coefficient, s.t. $\sum_i h_{zi} = 1$. R_i is the variational rate calculated from our method, but with no historical information, i.e., $\theta = 1$.

We use the proposed evolutionary clustering on this generated network series and got 25 overlapping communities in each snapshot. We measure the similarity at each timestamp between the fastest growing community detected by each method(**VACS**, *KNN*, *CID*) with community C_1 using Jaccard coefficient. On the other hand, as we notice from the way of community generation, these five communities increase their weights with higher probability on internal edges than outward edges, and they more or less increase their internal strength over time. Hence the community with weakest growing trend should consist of nodes well separated to these predefined communities. To measure this, we use entropy $S = -\sum_{i=1}^{5} P(x \in C_i) log(P(x \in C_i))$. We show the experimental results in Figure 3 and Table 2.

Table 2: Performance on Synthetic Dataset

Method	Average Jaccard Coefficient (Fastest Growing)	Average Entropy (Slowest Growing)
VACS	**0.6006**	**2.0658**
KNN	0.5468	1.8307
CID	0.4429	1.9114

Table 2 presents the average value in ten snapshots. We can observe that our method outperforms baselines in both fastest and slowest growing community detection. In Figure 3, we can also observe that, as the time passes, the value of Jaccard coefficient via our method increases more consistently with less perturbation. This well demonstrates the effect of temporal smoothness, as we not only consider current growth, but also combine the trend in past time. The overall growth in Figure 3(a) and overall decrease in Figure 3(b) also illustrates the better community detection result, which well demonstrates the effectiveness of evolutionary clustering.

4.2 Social Evolution Dataset

The social evolution dataset records the daily activities from students through their mobile phones. It was collected by MIT dynamics lab [21], and provides the information about the call, message and sharing in a given period. The dataset also gives information from surveys taken on *10/19/2008, 12/13/2008, 3/5/2009, 4/17/2009* and

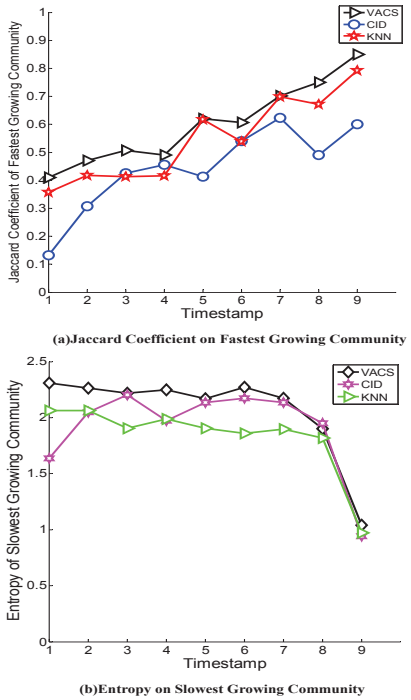


Figure 3: Performance on Synthetic Data.

Table 3: Weights for Various Friendship Categories

Friendship Level	Redefined	Weight
Blank	Do Not Know	0
Facebook All Tagged Photos	Not Familiar	1
Political Discuss Socialize Twice Per Week	Acquaintance	2
Close Friend	Friend	3

5/22/2009, respectively. To validate our result using this information, we divided the original interaction records into five snapshots by the date of surveys, and constructed network W^k from each snapshot. W^k is a symmetric matrix in which $(i,j)^{th}$ entry denotes the frequency of interactions between i^{th} student and j^{th} student. We also construct friendship network F^t. In each survey, the student was asked his friendship with others in terms of the five levels given in Table 3. We assign the weight on each edge from 0 to 3 based on the level of friendship.

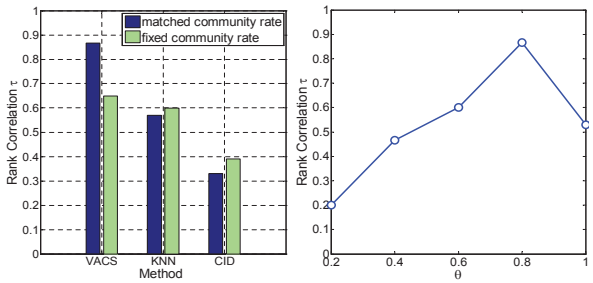
To validate the variation detection using this friendship network F^t, we first measure the intensity of each community in F^t by the gap between the inside-community average friendship and the outside-community average friendship. The community division is based on membership matrices $\{C^t\}_{t=1,..,T}$, which has already been decided from the interaction network by earlier mentioned evolutionary clustering procedure. The intensity value will be higher if students in same community know each other well, while they have few connections outward. After we gathered the intensity values, we calculate the friendship variational rate for each community at time t with two versions. In the first version we fix the community and the rate is computed as $Rate(z) = Intens_{t+1}(z)/Intens_t(z)$. The second version takes into consideration of clustering evolution and matching, and can be calculated by equation $Rate(z) =$

$M_{-1_z} * Intens_{t+1}/Intens_t(z)$. M_{-1} is the matching matrix from t to $t+1$, $Intens_t$ is the column vector of intensity value for all communities at time t. The corresponding CID baseline equation in second version will become $Rate_C(z) = M_{-1_z} * CID_{t+1}/CID_t(z)$ and the same modification goes for KNN baseline. For both cases, the higher rate value indicates the trend of higher intimacy for this group. Then we rank the variation rate for all communities at same timestamp, which we call rank L_f.

As this friendship information reflects the real belief by students, it can be used to validate the inference by our method from their daily activities. With the variational rate value figured out by the **VACS** algorithm, we rank the value in each snapshot as L_v. We then verify whether our results match the survey information, by measuring the correlation between L_f and L_v, using Kendall's τ coefficient [16]. The result is shown in Figure 4(a). Kendall's τ ranges from -1 to 1, where -1 indicates completely reverse ranking and 1 indicates identical ranking. The higher the value, the more correlated the two rankings are. It can be easily observed from Figure 4(a) that **VACS** outperforms baselines in both with and without community matching cases. As *CID* mainly focuses on internal density and ignores the outward edges, it may not be able to measure the friendship precisely. Compared with *CID*, *KNN* performs much better, as it uses a different community metric, and also involves local information. However, it did poorly compared with **VACS** as it does not take into consideration of the temporal information. On the other hand, **VACS** matches the real friendship information best, as it not only involves smoothness, but also reduces the noise. Even with the fixed community in first version, **VACS** with evolutionary view outperforms baselines, which are designed specifically for fixed membership. When we consider the evolution of communities, or use second version, the advantage of **VACS** becomes more obvious, despite the revision for baseline methods in evolutionary view.

In this dataset, the attitude of students will not change dramatically in short time. If two students i and j keep close communication for a long time, and in one snapshot their interaction frequency drops sharply, this is more likely to be caused by some personal reason rather than by the sudden change of great friends to strangers. On the contrary, if two students never know each other before, and from a certain timestamp t they start to message each other very frequently. This usually means they share some topics and get to know each other, but their belief of friendship will not change directly from strangers to close friends. In most cases, this change takes place gradually. Hence, temporal smoothness is very necessary in this case. The result well demonstrates **VACS** can reflect the variational trend of students' real belief on friendship.

We then evaluated the influence from historical information and the temporal outlier detection. Figure 4(b) shows the relation between rank correlation τ and θ. We can observe that it reaches its peak at around $\theta = 0.8$. This hill-shaped curve well illustrates that the best measure takes place with a good combination of current and historical information. When θ is close to one, it loses historical information and thus little smoothness is involved. When θ is very small, it works poorly. This is predictable as we cannot rely completely on historical trend while almost ignoring what happens currently. Figure 5 shows the interaction and



(a)Performance with Baselines (b)Sensitivity of Weight θ

Figure 4: Performance on Social Evolution Dataset.

friendship variational rate of temporal outlier communities, detected with largest a_k value at each snapshot. Interaction variational rate on community z at time t is defined as $InterRate(z) = NorStrength(z)/NorHStrength(z)$, and friendship variational rate is defined as $FriendRate(z) = Intens_t(z)/Intens_{t-1}(z)$, which represents the variational rate of real friendship in community. We can observe that large gap exists between the two curves when the community is identified as a temporal outlier. Hence the variational rate estimated from interaction network cannot truly reflect the friendship variational trend for these detected communities, which well supports the effectiveness of our method.

We also validated the effectiveness of evolutionary clustering in our framework. By utilizing the friendship network information, we calculate the clustering quality for the interaction networks. Similar to the previous experiments, for each community we use the gap between average internal friendship and outward friendship weights as the quality measure. Then we take the average for all detected communities as the overall clustering quality. We make comparison on the results with different historical penalty weights, as well as hard clustering baseline. The result is shown in Figure 6. As in this case one student may belong to multiple friendship communities, a hard clustering method is less meaningful than soft correspondence. We can observe that the hard clustering method K-means performs poorly compared with the soft NMF-based method. At the first timestamp, we have no historical information and thus the quality value is very close for curves with different weights. It can be observed that the clustering quality is optimal when α is around 0.15. At this point it reaches the best tradeoff between historical and current information. After we combine the historical information, we can find the performance will become better over time. When α reaches 1, which means we rely 50% on history, we get a relatively poor clustering. Thus, we conclude that the temporal smoothness is necessary, while the weight for combination should be chosen properly.

4.3 DBLP Dataset

We applied our method on a subset of DBLP Dataset [28]. There are totally 144,179 papers involved which are published during 1991-2000. We divided them into 5 snapshots, each with a 2 year period. There are in total 1059 authors and 43 venues involves. From the provided information, we construct an author-to-author network from each timestamp and set the weight for each edge as the collaboration times during that period. Two nodes will not be linked if the corresponding authors share no overlap on any publication.

Table 4: Authors and Publication in Significantly Growing Group

DBLP Dataset	1991-1992	1993-1994	1995-1996	1997-1998	1999-2000
Author	F.Esposito G.Semeraro D.Malerba R.Hennicker M.Wirsing	F.Esposito G.Semeraro D.Malerba R.Hennicker M.Wirsing	C.Tang,F.Esposito G.Semeraro D.Malerba,T.Hsu R.Hennicker S.Peng,M.Wirsing	C.Tang,F.Esposito G.Semeraro D.Malerba,T.Hsu R.Hennicker S.Peng,M.Wirsing M.Costabile	C.Tang,F.Esposito G.Semeraro D.Malerba,T.Hsu R.Hennicker S.Peng,M.Wirsing M.Costabile
Corresponding Venue	M.Programming(1) IPMU(1) PAMI(1)	AI*IA(1) ECML(2) Applied AI(1) ESOP(1) ISMIS(1) LOPSTR(1)	Sci.Comput.Program.(1) AI*IA(1) ECML(1) GULP-PRODE(1) ISAAC(1) ISMIS(1) LOPSTR(1)	Applied AI(2) Theor.Comput.Sci.(1) PAMI(1),FCT(1) AI*IA(3),ICDAR(1) ISMIS(1) ECDL(2),COCOON(2) Int.J.on Dig.Lib.(1) VDB(2),LOPSTR(1)	Int.J.on Dig.Lib.(1) UML(1),MLDM(1) Machine Learning(1) Theor.Comput.Sci.(1) J.Intell.Inf.Syst.(1) Algorithmica(1),ISMIS(5) ILP(1),AI*IA(2),ECDL(1) ICML(1),ISAAC(1)

Table 5: Actors and Collaboration Frequency in Significantly Growing Group

Actors	1991-1993	1994-1996	1997-1999	2000-2002
D.Dowse,D.Deluise,J.Jones,E.Murphy,M.McKean,K.Campbell J.Vargas,M.Rapaport,K.Wilson,J.Adams,P.Proctor,R.Sarafian J.Witherspoon,K.Pollak,C.Kusatsu,G.Gress	4	5	7	128
L.Lombardi,F.Medrano,C.Rocket,P.Herman,J.McGinley B.Crystal,S.Wilson,M.Papajohn,T.Rosales Jr.	1	2	12	11
J.Cusack,A.Arkin,J.Roberts,C.Walken,B.Crystal,J.Cusack,D.Krumholtz B.Pepper,D.Aykroyd,S.Tucci,A.Cusack,L.King,J.Gandolfini,G.Lewis	4	2	20	44

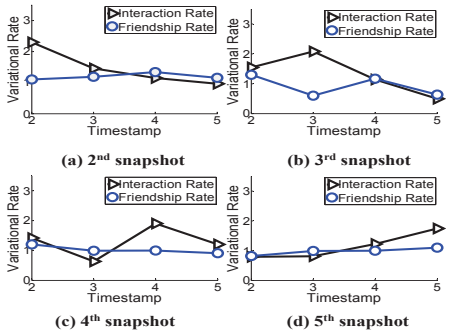


Figure 5: The Temporal Outlier Detection.

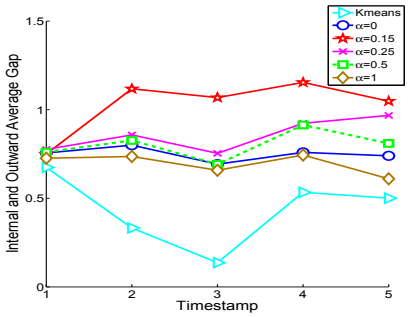


Figure 6: The Influence of Historical Information.

We detect the author communities that grow significantly in collaboration. We first detect the most consistently growing community from 1991-2000 by the method mentioned above. We assume it to be community z that belongs to time t. In this dataset we hold the assumption that the community evolves with relative strong internal intensity. Hence it is less likely to find dramatic membership change during evolution, such as the case with one community equally split to multiple small communities. Based on this assumption, we then find from each snapshot other than t the community that matches z best. Table 4 shows the fastest growing community and the corresponding conferences or journals where they coauthored. The number enclosed in parenthesis denotes the times of collaboration in this venue.

We can easily observe from Table 4 that the collaboration frequency for this group keeps growing consistently, from only 3 times in the first snapshot to 18 times in the last snapshot. We can also find that new members keep joining as time progresses, who started collaboration with authors already in this group, from which we can infer that the field they are working on is quite attractive during the period. By observing the venues of their publications, we can conclude that these authors work in relatively close and limited areas. Further, they started to coauthored more consistently over time in certain conferences or journals such as AI*IA and ISMIS. This well explains their high variational rate and shows the effectiveness of our method.

4.4 IMDB Dataset

In this experiment we detect the top fastest growing movie star communities by applying our method on a subset of IMDB dataset[1]. This dataset involves the American made comedy movies and TV plays from *1991-2002*, together with the information of corresponding actors that participate in them. We construct an actor-to-actor network from four snapshots, each with a 4 year period. The weight of each edge (v_i, v_j) will be set as the collaboration frequency between actor i and actor j during that period. Totally 700 movie stars are involved in this dataset.

[1]www.imdb.com/interfaces

We run the **VACS** algorithm to measure the variational rate for star communities. We detect the top-3 fastest growing communities and track the collaboration for each of them from 1991 to 2002. The detailed information is given in Table 5. The 2nd to 5th columns in the table indicates the times of cooperation inside this group during each time period. We can observe that all three communities show the rapid growth trend. The collaboration of the third community decreased at the beginning, but later shows a strong rebound, which makes it an overall top-3 fastest growing group during 1991-2002. All these three communities hold the similarity that the actors in each one share similar interest in certain type of movie, or they have established consistent collaboration on some series movie or TV plays. The results well support that the communities detected by **VACS** are really meaningful.

5. CONCLUSIONS

In this paper, we proposed a method called **VACS** to analyze the variational rate for communities on dynamic networks. The **VACS** algorithm involves information from clustering evolution, temporal smoothness and noise reduction. The whole framework can be implemented by two separate stages. At the first stage we conducted evolutionary clustering, soft correspondence matching and temporal strength calculation. At the second stage we solved the proposed optimization model and obtained the estimation of the variational rate for each community. The effectiveness is well supported by the validation on synthetic and social evolution datasets, in which our method outperforms baselines in various comparisons. The case studies on DBLP and IMDB datasets also show very impressive results with interesting information uncovered from the detected significant communities.

6. ACKNOWLEDGEMENT

The materials published in this paper are partially supported by the National Science Foundation under Grants No. 1218393 and No. 1016929.

7. REFERENCES

[1] B. Abrahao, S. Soundarajan, J. Hopcroft, and R. Kleinberg. On the separability of structural classes of communities. In *Proceedings of the 18th ACM SIGKDD*, pages 624–632. ACM, 2012.

[2] 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*, pages 44–54. ACM, 2006.

[3] T. Berger-Wolf, C. Tantipathananandh, and D. Kempe. Dynamic community identification. *Link Mining: Models, Algorithms, and Applications: Models, Algorithms, and Applications*, page 307, 2010.

[4] S. Bhagat, G. Cormode, B. Krishnamurthy, and D. Srivastava. Privacy in dynamic social networks. In *Proceedings of the 19th WWW*, pages 1059–1060. ACM, 2010.

[5] D. Chakrabarti, R. Kumar, and A. Tomkins. Evolutionary clustering. In *Proceedings of the 12th ACM SIGKDD*, pages 554–560. ACM, 2006.

[6] H. Cheng, Y. Zhou, X. Huang, and J. X. Yu. Clustering large attributed information networks: an efficient incremental computing approach. *Data Mining and Knowledge Discovery*, 25(3):450–477, 2012.

[7] Y. Chi, X. Song, D. Zhou, K. Hino, and B. L. Tseng. Evolutionary spectral clustering by incorporating temporal smoothness. In *Proceedings of the 13th ACM SIGKDD*, pages 153–162. ACM, 2007.

[8] Y. Chi, X. Song, D. Zhou, K. Hino, and B. L. Tseng. On evolutionary spectral clustering. *ACM TKDD*, 3(4):17, 2009.

[9] J. Cook, K. Kenthapadi, and N. Mishra. Group chats on twitter. In *WWW 2013*, pages 225–236. ACM, 2013.

[10] A. L. Creekmore, W. T. Silkworth, D. Cimini, R. V. Jensen, P. C. Roberts, and E. M. Schmelz. Changes in gene expression and cellular architecture in an ovarian cancer progression model. *PloS one*, 6(3):e17676, 2011.

[11] X. H. Dang and J. Bailey. A framework to uncover multiple alternative clusterings. *Machine Learning*, pages 1–24, 2013.

[12] C. Ding, T. Li, W. Peng, and H. Park. Orthogonal nonnegative matrix t-factorizations for clustering. In *Proceedings of the 12th ACM SIGKDD*, pages 126–135. ACM, 2006.

[13] N. Du, J. Gao, and A. Zhang. Progression analysis of community strengths in dynamic networks. In *ICDM 2013*. IEEE, 2013.

[14] M. Gupta, J. Gao, Y. Sun, and J. Han. Integrating community matching and outlier detection for mining evolutionary community outliers. In *Proceedings of the 18th ACM SIGKDD*, pages 859–867. ACM, 2012.

[15] M. Jakobsson and N. A. Rosenberg. Clumpp: a cluster matching and permutation program for dealing with label switching and multimodality in analysis of population structure. *Bioinformatics*, 23(14):1801–1806, 2007.

[16] M. G. Kendall. A new measure of rank correlation. *Biometrika*, 30(1/2):81–93, 1938.

[17] J. Leskovec, K. J. Lang, and M. Mahoney. Empirical comparison of algorithms for network community detection. In *WWW 2010*, pages 631–640. ACM, 2010.

[18] Y.-R. Lin, Y. Chi, S. Zhu, H. Sundaram, and B. L. Tseng. Analyzing communities and their evolutions in dynamic social networks. *ACM TKDD*, 3(2):8, 2009.

[19] C. Liu, H.-c. Yang, J. Fan, L.-W. He, and Y.-M. Wang. Distributed nonnegative matrix factorization for web-scale dyadic data analysis on mapreduce. In *Proceedings of the 19th WWW*, pages 681–690. ACM, 2010.

[20] B. Long, Z. Zhang, and P. S. Yu. Combining multiple clusterings by soft correspondence. In *ICDM 2005*, pages 8–pp. IEEE, 2005.

[21] A. Madan, M. Cebrian, S. Moturu, K. Farrahi, and S. Pentland. Sensing the ąőhealth stateąŕof a community. *IEEE Pervasive Computing*, 11(4):36–45, 2012.

[22] S. Mankad and G. Michailidis. Structural and functional discovery in dynamic networks with non-negative matrix factorization. *Physical Review E*, 88(4):042812, 2013.

[23] Y. Mehmood, N. Barbieri, F. Bonchi, and A. Ukkonen. Csi: Community-level social influence analysis. In *Machine Learning and Knowledge Discovery in Databases*, pages 48–63. Springer, 2013.

[24] E. E. Papalexakis, L. Akoglu, and D. Ience. Do more views of a graph help? community detection and clustering in multi-graphs. In *Information Fusion (FUSION)*, pages 899–905. IEEE, 2013.

[25] D. Saez-Trumper, G. Comarela, V. Almeida, R. Baeza-Yates, and F. Benevenuto. Finding trendsetters in information networks. In *ACM SIGKDD*, 2012.

[26] K. Subbian, C. C. Aggarwal, J. Srivastava, and P. S. Yu. Community detection with prior knowledge. In *SDM*, pages 405–413, 2013.

[27] J. Sun, D. Tao, and C. Faloutsos. Beyond streams and graphs: dynamic tensor analysis. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 374–383. ACM, 2006.

[28] L. Tang, H. Liu, J. Zhang, and Z. Nazeri. Community evolution in dynamic multi-mode networks. In *Proceedings of the 14th ACM SIGKDD*, pages 677–685. ACM, 2008.