Dynamic Community Detection with Normal Distribution in Temporal Social Networks

Yaowei Huang
5140309539
14330222150355@sjtu.edu.cn
Shanghai Jiao Tong University

Yuchen Lin
5140309507
yuchenlin@sjtu.edu.cn
Shanghai Jiao Tong University

Zhaozhe Song
5140309514
songzhaozhe@126.com
Shanghai Jiao Tong University

Supervisor: Prof. Luoyi Fu

***Abstract*—A community in networks is thought as a cluster of nodes with more connections amongst its members than between its members and outside nodes. Communities often have overlap since it is possible that some nodes belong to multiple communities. Overlapping community detection in static networks is to discover communities without considering the temporal evolution of the connections between nodes. However, it is evident that human behaviors in social networks are highly dynamic, which means that the relationship between nodes and communities changes with time going by. Due to the difficulties in evaluating the detected result and in incorporating temporal factors to the objective function, overlapping community detection in temporal social networks is still an open problem.**

In this paper, we present our direct dynamic community detection approach for temporal social networks, which is based on modeling distributions of the strength of each membership over time. The intuition behind our approach is based on an observation that most memberships of nodes to communities are either one-off within a short term or long term. Experimental results show that our approach achieves substantial improvement.

Keywords: community detection, temporal networks, dynamics of social interaction

I. Introduction

In a network, it is common that there are some groups of nodes, where the connections within groups are evidently denser than the connections crossing groups. Such groups are thought as *communities* in networks. We often allow that nodes can have memberships of multiple communities in social networks, and such communities are called "overlapping communities".[1] Community can be considered as a group of people frequently contacting with each other or sharing similar interests in social networks, a scientific discipline in citation networks and collaboration networks[2], a functional unit in biomedical networks, etc. Representing the basic structures of networks, community detection is essential to understand the organization of real-world networks, infer the relationships among nodes and better understand the network dynamics [1].

Static and Dynamic Networks Static networks have no temporal factors on nodes or edges and thus the relationship between nodes and detected communities is fixed. While a dynamic network incorporates the temporal information about interactions between nodes, so that we can obtain a dynamic detected communities. Many social networks can be viewed as both static and dynamic, depending on whether we utilize the temporal data [2]. Assume we have many tuples in the dataset like (u, v, t_u, t_v) denoting that a researcher u published a paper at time t_u cites a paper of another researcher v which is published at time t_v. We call such a tuple as an *interaction* in general networks. Static networks only contains information about node pairs (u, v), while dynamic networks take temporal factors t_u and t_v into consideration as well.

Static and Dynamic Community Detection It is true that static networks are simple to model for they prune much temporal information [3]. However, it is sometimes too simplified to discover the time varying relationship between nodes and communities especially in temporal social networks. Human behaviors are highly dynamic and consequently the relationship between nodes and communities are not independent

[1]We will always refer "community" to "overlapping community" in the following content.

[2]Citation networks and collaboration networks are thought as two kinds of social networks.

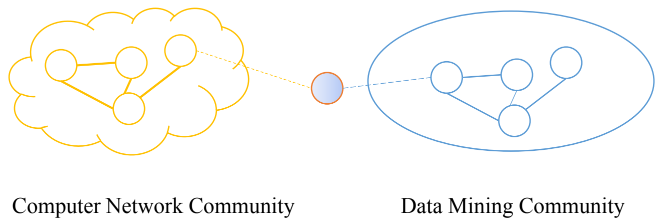

Fig. 1: A static network: the researcher (node) is fixed in between Computer Network community and Data Mining community.

on time changing. For example, if we have two communities in a certain citation network, which are about Computer Network and Data Mining respectively. It is possible that there is a node in this network denoting a researcher who had been doing research in Computer Network when he/she was a master student, but changed his/her research interests to data mining when he/she became a PhD student later. See Figure 1 and Figure 2. [3] Thus, this researcher was an active member of the Computer Network community but the strength of this membership decreases over time; while, the strength of his/her membership of Data Mining community increases with time going by.

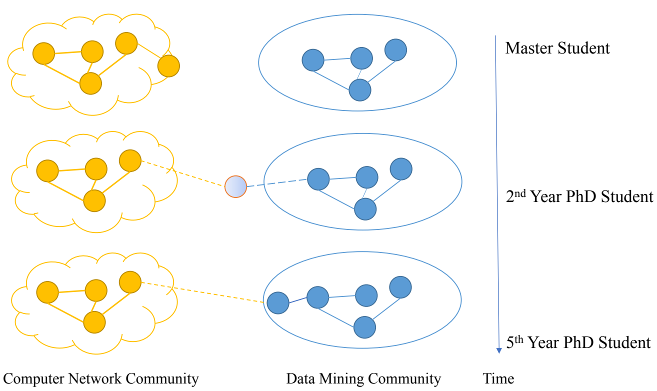

Fig. 2: A dynamic network: the researcher is an active member of Computer Network community when he/she was a master student; a less active researcher in both two community when he/she was a 2^{nd}-year PhD student; an active member of Data Mining community and an inactive member of Computer Network Community when he/she was a 5^{th} PhD student.

Due to many such potential temporal factors influencing the community structure of our networks, it is necessary to represent temporal factors in the strength of membership. Simply put, in static community detection we consider the strength of the membership between a node u and a community k as a function $f(u,k)$, while in *dynamic community detection* we consider it as a function $f(u,k,t)$ where t is a certain timestamp.

Present work: Static Community Detection Much work has been devoted to detecting communities in very complex networks in various disciplines such as computer science, statistics, applied mathematics, bioinformatics, social sciences, etc. Traditionally, the emergence of communities in networks has been understood through the strength-of-weak-ties theory [4], [5], which conceptualize networks as consisting of dense clusters that are linked by a small number of weak ties. Assuming this view of network communities, graph partitioning [6]–[8], modularity optimization ([9], [10]), and betweenness centrality based approaches for community detection concentrate on identifying edges which can be cut to separate the network into a set of non-overlapping clusters. However, in social networks and many other types of networks, overlapping community structure are more desired. Some overlapping community detection methods [11], [12] assume that the community overlaps are less densely connected than the non-overlapping part of communities. This unnatural assumption leads to either identify a dense overlap as a separate community or merge two overlapping communities into a single one.

BIGCLAM [11] model finds an increasing relationship between the number of shared communities and the probability of nodes being connected by an edge and then formulates community detection as a variant of non-negative matrix factorization (NMF) and then optimize the model likelihood of explaining the links of the observed network, which is very provisional.

Present work: Dynamic Community Detection Present work dynamic (overlapping) community detection can be divided into two kinds according to the information they leverage: *Indirect methods* and *Direct methods*: 1) Indirect methods firstly focus on identifying communities within a set of *snapshots* of the target network, and then synthesize a final model of their lifetime based on the time steps [7], [13]–[15]. In each snapshot of the network, temporal factors are thought as the same and thus they can utilize static community detection methods; 2) Direct methods design models with temporal factors to directly leverage the whole information in time series and then they can learn the community structure from the original dynamic network [16]–[18], Most of direct methods are based on tensor decomposition techniques, resulting low physical interpretation. We would like to discuss them more in

[3] Note that this is just a simple example to illustrate the dynamics, where only one node is moving over time.

detail in Section II.

In this paper, we present our direct dynamic (overlapping) community detection approach for temporal social networks, which is based on modeling distributions of the strength of each membership over time (Section IV and Section V). The intuition behind our approach is based on an observation that most memberships of nodes to communities are either one-off within a short term or long term (Section III). Experimental results in Section VI show that our approach achieves substantial improvement.

II. Related Work

Indirect dynamic community detection methods usually involves many clusters generated from all the snapshots. Previous work such as [15] only maintains those most frequently appearing ones. Evolutionary clustering (EC) [14] applies K-means clustering to a similarity matrix generated from the current snapshot and the clustering result of previous snapshots. Researchers have replaced the K-means clustering with other clustering methods or proposed new ways to generate similarity matrices [7], [8], [13], [19], [20] to improve EC. However, these methods use local information (more specifically, a similarity matrix generated from a small subset of snapshots) to identify clusters. When snapshots are sparse and contain too few edges, the similarity matrices provide little information for clustering, and the detected clusters can be small and fail to correspond to meaningful clusters. Moreover, nodes are usually assigned to only one cluster in a time step, while in real-world social networks or email networks, a person may be part of multiple communities at the same time.

Evolving stochastic block models (SBM) [5], [16], [17] are proposed for a dynamic network where each node has a mixed membership of communities defined by the model. A probability model is applied to learn the model. However, since these methods focus on computing the memberships, the formation, dissolution and lifetime of a community remains unknown. Tensor decomposition based methods such as [3], [18] model a network as a three mode tensor and apply low-rank tensor factorization to obtain R components. Each component consists of three vectors named loading vectors. Two of the loading vectors relate to nodes and are used to generate a community with binary classification. The other loading vector contains temporal information for tracking the community lifetime. However, previous analysis with tensor decomposition fails to provide a good model for the dynamic network; more precisely, the physical interpretations of the vectors related to nodes and the time are unclear. As a result, they inaccurately determine the lifetime of a community when network snapshots are sparse and contain few edges. They also fail to provide a way to accurately calculate the lifetime of communities. Some methods merge snapshots to analyze data at a large time granularity, but this can result in inaccurate lifetime detection for a cluster because of the loss in details of the change in a cluster.

III. Observation and Intuition

In the DBLP dataset we used, there are in total 71195 users, and 18638580 edges, i.e. (u,v) pairs, exist between them. Each edge consists of several time pairs (t_1, t_2) too. To analyze the data, we did some statistical analysis on the dataset.

In the dataset, each user has his/her ground truth area of research. However, the ground truth fails to capture the temporal evolution of users. To come up with a better ground truth for temporal distribution between users, we define our ground truth under the temporal scale in the following way.

For user u and each research area C_i, we count his/her number of interactions in the data with the users $v \in C_i$. We count the total number of interactions for u in each year. Using these numbers as the Y-axis and the time dimension as the X-axis, we can get a curve as a ground truth curve. We denote this curve as Cur. To some extent, because a user's community is reflected by the frequency he/she interacts with the people in the community, and the majority of the people's fields in every area are quite stable, the curve can reflect every user's temporal evolution of his area of research.

We also counted the percentage of people that their research area changed in the dataset. We defined a threshold σ for the variance of the curve Cur. Intuitively, if the variance is larger than σ, we will regard this user changed his research interest in a community. When using σ as 0.5, we find that 17% of the users in our dataset changed their interest.

These observations imply that building a model for the temporal evolution of communities, i.e., dynamic community detection, is highly essential to analyze temporal networks.

IV. Modeling

In this part we present the framework and generative process of our model $\boldsymbol{CDOT}$ (Community Detection Model of Multi-interaction Over Time), which

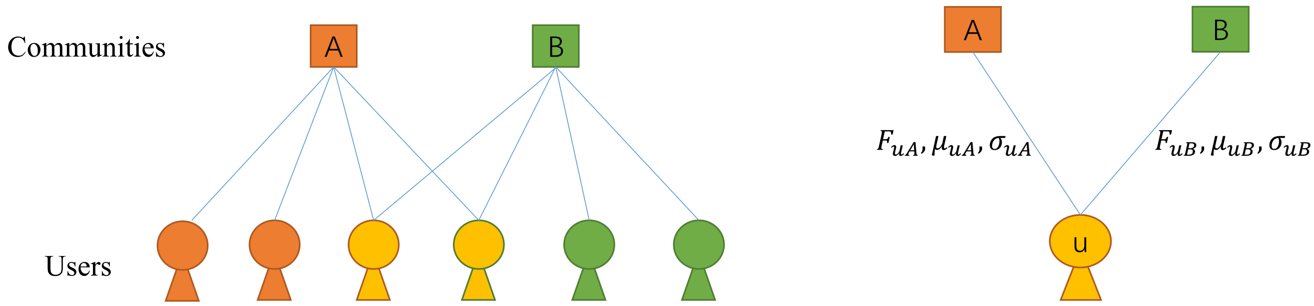

Fig. 3: Bipartite affiliation graph. Each user will connect communities with a weight like F_{uA} and a Guassian distribution with mean value μ_{uA} and variance σ_{uA}.

can observe the time information between users and communities. Fig. 3 illustrates our model. Given a bipartite graph where the nodes at one side represent the nodes of the social network G, the nodes on the other side represent communities C, and the edge has three parameters, F, μ, σ, the weight of the affiliation, mean value and variance of Guassian time distribution respectively.

A. Model Description

CDOT is based on the idea that communities arise due to shared group affiliation, and views the whole network as a result generated by a variant of the community-affiliation graph model. Same as the origin one, CDOT models the community affiliation strength between each pair of node u and community c with a nonnegative parameter $F_{uc}(F_{uc} = 0$ means node u is definitely not affiliated to community c.) However, CDOT also aim to detect $P_{uc}(t)$ which is a Guassian distribution between node u and community c over time t, so we consider that $F_{uc}P_{uc}$ is the community affiliation strength between node u and community c at time t.

We consider the input interaction network as a graph G(V,E) where V is a set of U users and E is a set of edges associated with two time stamps t_1 and t_2.

A directed edge (u, t_1, v, t_2) means the interaction between user u and v and t_1 belongs to u and t_2 belongs to v, and there may be multiple edges between a pair of nodes, which means there are several interactions between them. Take dblp as an example, if one person u publish a paper at time t_1 and the person v cites it at time t_2, then they will have one edge between them.

To generate a link $(u(t_1), v(t_2))$ with probability $p(u, t_1, v, t_2)$, we define that:

$$\begin{aligned} p(u,v,t_1,t_2) &= 1 - \exp(-\sum_{c=0}^{K} F_{uc}P_{uc}(t_1)F_{vc}P_{vc}(t_2)) \\ P_{uc}(t_1) &= \frac{1}{\sqrt{2\pi}\sigma_{uc}} \exp(-\frac{(t_1-\mu_{uc})^2}{2\sigma_{uc}^2}) \\ H &= \sum_{c=0}^{K} F_{uc}P_{uc}(t_1)F_{vc}P_{vc}(t_2) \end{aligned} \tag{1}$$

The process in Eq.1 suggests the possibility of node u in the time t_1 connects node v in the time t_2 within each community. There will be three parameter between node u and community c: $F_{uc}, \mu_{uc}, \sigma_{uc}$, where F_{uc} denotes the total strength between them, μ_{uc} and σ_{uc} are the mean and variance value of Gaussian distribution, where μ_{uc} denotes the maximum likelihood time of node u in the community c, and σ_{uc} evaluate if the person is "temporal" to the community or "stable", which means that if $\sigma_{uc} \leq \delta$ (δ is the threshold of temporal and stable), he is "temporal", belongs to the community at a special short period of time near μ_{uc}. If $\sigma_{uc} > \delta$, he is "stable", belongs to the community over a long time.

δ-Threshold. As shown in Fig. 4. In the above we have defined a threshold of σ determining if one node belongs to a community "temporal" or "stable", we find that if the σ_{uc} satisfy the function that $P_{uc}(\mu_{uc})/P_{uc}(|\mu_{uc} - 5|) > 0.5$ it is "temporal", otherwise "stable".

B. Community Detection Over Time

Now we have defined the CDOT model, we explain how to detect network communities using the model. Given the interaction network G(V,E), we aim to detect K communities by fitting the CDOT to the underlying network G by maximizing the log likelihood

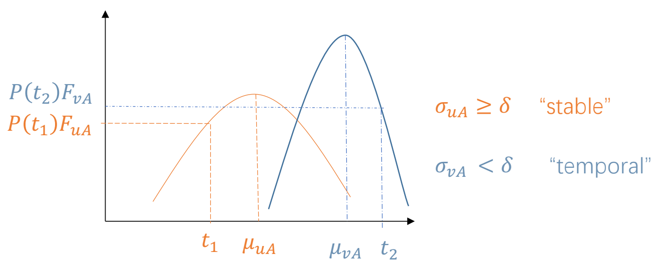

Fig. 4: $\sigma_{uA} \geq \delta$: the node u is "stable" to community A. $\sigma_{vA}<\delta$: the node v is "temporal" to community A.

$l(F, \mu, \sigma) = \log P(G|F, \mu, \sigma)$:

$$\hat{F}, \hat{\mu}, \hat{\sigma} = \underset{F \geq 0, \sigma > 0}{\arg\max} l(F, \mu, \sigma) \tag{2}$$

where

$$l(F, \mu, \sigma) = \sum_{(u,v,t_1,t_2) \in E} log(1 - \exp(-H)) - \sum_{(u,v,t_1,t_2) \notin E} H \tag{3}$$

where H is defined in E.q(1) As a matter of fact, we will notice that the second term here is infinity on number, because the time t is a continuous parameter, and the time t_1, t_2 is infinity. We decide to using sampling method to deal with this problem, sample some negative edges to be the second term. The detail of implementing will be explained in the next section.

V. Parameter Learning

To solve the optimization problem defined in Eq. 2, we adopt a block coordinate gradient ascent. We first update the community affiliation strength $\boldsymbol{F_u}$ for each node with other F_v and μ, σ fixed, and then update the mean value of Gaussian distribution in community affiliation $\boldsymbol{\mu_{uc}}$ for each node with other μ_{vc} and F, σ fixed, then the variance of Gaussian distribution in community affiliation $\boldsymbol{\sigma_{uc}}$ for each node with other σ_{vc} and F, μ fixed, because if we fix the other parameter, then the problem of updating will become a convex optimization problem, we solve the following subproblem:

$$\underset{F \geq 0}{\arg\max} l(F_u) \tag{4}$$

where

$$l(F_u) = \sum_{(u,v,t_1,t_2) \in N(u)} \log(1 - \exp(-H)) - \sum_{(u,v,t_1,t_2) \in NegN(u)} H \tag{5}$$

where $N(u)$ is a set of edges connecting to node u, and $NegN(u)$ is a set of Negative edges that we sampled from the nonexistent edges in origin network G. The subproblem can be further solved by projected gradient ascent.

$$F_{uc}^{new} = \max(0, F_{uc}^{old} + \alpha_{F_u} \nabla(F_{uc})) \tag{6}$$

where α_{F_u} is the step size computed by backtracking line search, and the gradient is:

$$\begin{aligned} \nabla l(F_{uc}) = & \sum_{(v,t_2) \in N(u)} \frac{\exp(-H)}{1 - \exp(-H)} P_{uc}(t_1) P_{vc}(t_2) F_{vc} \\ & - \sum_{(v,t_2) \in NegN(u)} P_{uc}(t_1) F_{vc} P_{vc}(t_2) \end{aligned} \tag{7}$$

Negative edges. Here we will introduce the sampling method of Negative edges from the origin network G. We use uniform distribution to sample some nodes and then to sample some time to be the negative edges, the number of Negative edges will equal to a definite ratio of the origin edges. The first reason has mentioned above, the Negative edges is infinity because time t is a continuous distribution. The second reason is that in a huge network a node will only have some edges between several nodes which are close to it or having high weight in the same community, not all node. If you consider all node in one community (a big one), then the number of Negative edges will be too large, which will make the F_{uc} as small as possible, even to zero.

After the community affiliation matrix $\mathbf{F}$ updated, we fix $\mathbf{F}$ and $\boldsymbol{\sigma}$,update the community time parameter matrix $\boldsymbol{\mu}$:

$$\mu_{uc}^{new} = \max(0, \mu_{uc}^{old} + \alpha_{\mu_u} \nabla(\mu_{uc})) \tag{8}$$

where α_{μ_u} is computed as α_{F_u}, and the gradient is:

$$\begin{aligned} \nabla l(\mu_{uc}) = & \sum_{(v,t_2) \in N(u)} \frac{\exp(-H)}{1 - \exp(-H)} F_{uc} P_{vc}(t_2) F_{vc} P_{uc}(t_1) \frac{t_1 - \mu_{uc}}{\sigma_{uc}^2} \\ & - \sum_{(v,t_2) \in NegN(u)} F_{uc} P_{uc}(t_1) F_{vc} P_{vc}(t_2) \frac{t_1 - \mu_{uc}}{\sigma_{uc}^2} \end{aligned} \tag{9}$$

After updating the community time parameter matrix $\boldsymbol{\mu}$, we fix $\mathbf{F}$ and $\boldsymbol{\mu}$, update the community time parameter matrix $\boldsymbol{\sigma}$:

$$\sigma_{uc}^{new} = \max(0, \sigma_{uc}^{old} + \alpha_{\sigma_u} \nabla(\sigma_{uc})) \tag{10}$$

where

$$\begin{aligned} \nabla l(\sigma_{uc}) = & \sum_{(v,t_2) \in N(u)} \frac{\exp(-H)}{1 - \exp(-H)} F_{uc} P_{vc}(t_2) F_{vc} P_{uc}(t_1) \frac{(t_1 - t_{uc})^2 - \sigma_{uc}^2}{\sigma_{uc}^3} \\ & - \sum_{(v,t_2) \in NegN(u)} F_{uc} P_{uc}(t_1) F_{vc} P_{vc}(t_2) \frac{(t_1 - t_{uc})^2 - \sigma_{uc}^2}{\sigma_{uc}^3} \end{aligned} \tag{11}$$

Model Initialization We just use the result of BIGCLAM to initialize $\mathbf{F}$, and for each node u and community c whose $F_{uc}>0$ initialize μ_{uc} (the midpoint of the whole period) and $\sigma_{uc} = 1$.

Determining community membership. To find the number of communities K, we need to determine if the node belongs to a community. For a "stable" user, we consider it belongs to the community. For a "temporal" user, we add a threshold δ_k that if $F_{uc}P(\mu_{uc})>\delta_k$ we could consider the node u belongs to community c, where

$$\delta_k = \sqrt{-\log\left(1 - \frac{2|E|}{|V|(|V|-1)}\right)} \qquad (12)$$

and the threshold is derived from the assumption that if two nodes belong to the same community k, then the probability of having an link between them through community k is larger than the backgound edge probability.

Choose the number of communities. We use the method to choose the number of communities K. And we reserve 20% of links for validation and learn the model parameters with the remaining 80%. The whole process is included in Algorithm 1.

Algorithm 1 Parameter Learning for CDOT

Input: $G(V, E; T)$: the temporal interaction network; max_iter: maximum number of iterations;
Output: **F**: the community affiliation;
$\boldsymbol{\mu}$: the mean of Gaussian distribution parameter matrix;
$\boldsymbol{\sigma}$: the variance of Gaussian distribution parameter matrix;

```
Determine the number of communities K.
Initialize F, μ, σ.
Calculate ∇l(μ_u) based on Eq.(9)
Calculate αμ_u using backtracking line search
Update μ_u based on Eq.(8)
Calculate ∇l(σ_u) based on Eq.(11)
Calculate ασ_u using backtracking line search
Update σ_u based on Eq.(10)
repeat
    Sample Negative edges in a definite ratio of origin edges
    for u = 1, 2, ..., N do
        Calculate ∇l(F_u) based on Eq.(7)
        Calculate αF_u using backtracking line search
        Update F_u based on Eq.(6)
    for u = 1, 2, ..., N do
        Calculate ∇l(μ_u) based on Eq.(9)
        Calculate αμ_u using backtracking line search
        Update μ_u based on Eq.(8)
    for u = 1, 2, ..., N do
        Calculate ∇l(σ_u) based on Eq.(11)
        Calculate ασ_u using backtracking line search
        Update σ_u based on Eq.(10)
until convergence or max_iter is reached
Return parameters F, t, σ.
```

VI. Evaluation

We proceed by evaluating the performance of the CDOT model. We evaluate our model from a number of different aspects.

A. *Convergence of CDOT*

1) Scalability: We evaluate the scalability of CDOT by measuring the running time on the networks of increasing sizes. We iterate our fitting process until $l(F, \mu, \sigma)$ fails to improve much in the last 3 iterations.

2) Quality of solution: The model in our formulation is not convex, meaning it might not converge to an optimal solution. To verify that our fitting algorithm does not suffer too much from local optima, we conduct the following experiment on synthetic networks. We generated 100 synthetic networks. For each of these networks, we then fit CDOT using 10 different random starting points and attempt to recover the true community affiliations.

B. *Accuracy of Community Detection*

In this section, we ignore the time evolution, and investigate the values of F_{uc}, $u \in V$ and $c \in C$. The reason why we can ignore time evolution here is that due to the fact that F_{uc} plays the major role in determining the extent to which a user belongs to a community, because the time dimension will always integrate to 1, having no influence on the overall strength.

We have the ground truth communities for each user. In particular, in our dataset of scholars, we know which areas of research each scholar belongs to. This allows us to quantify the accuracy of community detection methods by evaluating the level of correspondence between detected and ground-truth communities.

1) Experimental Setup: We are given an unlabeled undirected net- work G (with known ground-truth communities C^*) we aim to discover communities $\hat{C}$ such that discovered communities $\hat{C}$ closely match the ground-truth communities C^*.

Even though our algorithm can process the large network dataset, all the baseline methods do not scale to networks of such size. To allow for comparison between our and the baseline methods we use the following evaluation scenario where the goal is to obtain a large set of relatively small subnetworks with overlapping community structure. To obtain one such subnetwork we pick a random node u in the given graph G that belongs to at least two communities. We then take the subnetwork to be the induced subgraph of G consisting of all the nodes that share at least one ground-truth community membership with u. In our experiments we created 500 different subnetworks for each of the six datasets.

2) Baselines: For baselines we choose three most prominent overlapping community detection methods:

Link clustering (LC) [21], Clique Percolation Method (CPM) [22], and the Mixed-Membership Stochastic Block Model (MMSB) [23]. These methods have a number of parameters that need to be set. For CPM, we set the clique size k = 5 since the number of communities discovered by CPM with k = 5 best approximates the true number of communities. For MMSB, we have to set the number of communities K as an input parameter. We use the Bayes Information Criterion to choose K. While we require hard community memberships, MMSB returns stochastic node memberships to each of the K communities. Thus, we assign a node to a community if the corresponding stochastic membership is non-zero. We also considered Infomap [24], which is the-state-of-the-art non-overlapping community detection method. We omit the results as the performance of the method was not competitive.

3) Evaluation Metrics: The availability of ground-truth communities allows us to quantitatively evaluate the performance of community detection algorithm. Without ground-truth such evaluation is simply not possible. For evaluation, we use metrics that quantify the level of correspondence between the detected and the ground-truth communities. Given a network G(V, E), we consider a set of ground truth communities C^* and a set of detected communities $\hat{C}$ where each ground-truth community $C_i \in C^*$ and each detected community $C_i \in \hat{C}$ is defined by a set of its member nodes. To quantify the level of correspondence of $\hat{C}$ to C^* we consider:

1) Average F1 score. To compute the F1 score, we need to determine which $C_i \in C^*$ corresponds to which $\hat{C}_i \in \hat{C}$. We define F1 score to be the average of the F1-score of the best-matching ground-truth community to each detected community, and the F1-score of the best-matching detected community to each ground-truth community:

$$\frac{1}{2}\left(\frac{1}{|C^*|}\sum_{C_i \in C^*} F1(C_i, \hat{C_{g(i)}}) + \frac{1}{|\hat{C}|}\sum_{\hat{C}_i \in \hat{C}} F1(\hat{C_{g(i)}}, C_i)\right) \tag{13}$$

where the best matching g and g' is defined as follows:

$$g(i) = \arg\min_j F1(C_i, \hat{C}_j) \tag{14}$$

$$g'(i) = \arg\min_j F1(C_j, \hat{C}_i) \tag{15}$$

and $F1(C_i, \hat{C_{g(i)}})$ is the harmonic mean of Precision and Recall.

2) Omega Index [25] is the accuracy on estimating the number of communities that each pair of nodes shares:

$$\frac{1}{|V|^2}\sum_{u,v \in V} 1\{|C_{uv}| = |\hat{C_{uv}}|\} \tag{16}$$

where C_{uv} is the set of ground-truth communities that u and v share and $\hat{C_{uv}}$ is the set of detected communities that they share.

3) Normalized Mutual Information adopts the criterion used in information theory to compare the detected communities and the ground-truth communities. Normalized Mutual Information [26] has been proposed as a performance metric for community detection.

4) Accuracy in the number of communities is the relative accuracy between the detected and the true number of communities:

$$1 - \frac{||C^*| - |\hat{C}||}{2|C^*|} \tag{17}$$

For all metrics higher values mean more accurately detected communities, i.e. detected node community memberships better correspond to ground-truth node community memberships. Maximum value of 1 is obtained when the detected communities perfectly correspond to the ground-truth communities.

C. Accuracy of Time Modeling

In this section, we investigate the accuracy of the time model in our model. We first illustrate our evaluation metric, and then introduce our results.

1) Ground truth: The relationship of each user to each community is characterized by $F_{uc}, \sigma_{uc}, \mu_{uc}$. It is a Gaussian distribution. We will compare this distribution to the ground truth.

The ground truth is constructed in the following way. As shown in the previous section, for each community $\hat{C}_i$, we can map it to a ground truth community C_i. For user u, we count his/her number of interactions in the data with the users $v \in C_i$. We count the total number of interactions for u in each year. Thus, we can get a curve as a ground truth curve. We denote this curve as Cur.

2) Evaluation metric: We compare our Gaussian distribution curve with the ground truth curve Cur. To quantify this, we use the Pearson correlation coefficient.

VII. Conclusion

We propose a direct model with normal distribution to first represent the temporal factors that the strength of the membership in temporal social networks and then to detect dynamic communities. Both observation and experiment validate that the normal distribution is a great choice to approximate the real distribution. Our model increases only a very small number of parameters than other tensor decomposition based methods. While it achieves a substantial improvements than the baseline methods.

References

[1] R. Bro and H. A. L. Kiers, "A new efficient method for determining the number of components in parafac models," *Journal of Chemometrics*, vol. 17, no. 5, p. 274286, 2003.

[2] M. E. Newman, "Modularity and community structure in networks," in *APS March Meeting*, 2006, pp. 8577–8582.

[3] L. Gauvin, A. Panisson, and C. Cattuto, "Detecting the community structure and activity patterns of temporal networks: a non-negative tensor factorization approach," *Plos One*, vol. 9, no. 1, p. e86028, 2014.

[4] A. Y. Ng, M. I. Jordan, and Y. Weiss, "On spectral clustering: Analysis and an algorithm," *Proceedings of Advances in Neural Information Processing Systems*, vol. 14, pp. 849–856, 2002.

[5] T. Yang, Y. Chi, S. Zhu, Y. Gong, and R. Jin, "Detecting communities and their evolutions in dynamic social networksabayesian approach," *Machine Learning*, vol. 82, no. 2, pp. 157–189, 2011.

[6] R. Sands and F. W. Young, "Component models for three-way data: An alternating least squares algorithm with optimal scaling features," *Psychometrika*, vol. 45, no. 1, pp. 39–67, 1980.

[7] L. Tang, H. Liu, J. Zhang, and Z. Nazeri, "Community evolution in dynamic multi-mode networks," in *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2008, pp. 677–685.

[8] K. S. Xu, M. Kliger, and A. O. Hero, "Evolutionary spectral clustering with adaptive forgetting factor," in *IEEE International Conference on Acoustics, Speech and Signal Processing*, 2010, pp. 2174–2177.

[9] Y. Xu and W. Yin, "A block coordinate descent method for regularized multiconvex optimization with applications to nonnegative tensor factorization and completion," *Siam Journal on Imaging Sciences*, vol. 6, no. 3, pp. 1758–1789, 2013.

[10] E. Acar, D. M. Dunlavy, T. G. Kolda, and M. Mrup, "Scalable tensor factorizations with missing data," in *Siam International Conference on Data Mining, SDM 2010, April 29 - May 1, 2010, Columbus, Ohio, Usa*, 2010, pp. 701–712.

[11] J. Yang and J. Leskovec, "Overlapping community detection at scale:a nonnegative matrix factorization approach," in *ACM International Conference on Web Search and Data Mining*, 2013, pp. 587–596.

[12] E. Keogh, *SEGMENTING TIME SERIES: A SURVEY AND NOVEL APPROACH*, 2003.

[13] E. Acar, D. M. Dunlavy, and T. G. Kolda, "A scalable optimization approach for fitting canonical tensor decompositions," *Journal of Chemometrics*, vol. 25, no. 2, pp. 67–86, 2015.

[14] D. Chakrabarti, R. Kumar, and A. Tomkins, "Evolutionary clustering," in *Twelfth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Philadelphia, Pa, Usa, August*, 2011, pp. 554–560.

[15] Y. C. Chen, E. Rosensweig, J. Kurose, and D. Towsley, "Group detection in mobility traces," in *International Wireless Communications and Mobile Computing Conference, Iwcmc 2010, Caen, France, June 28 - July*, 2010, pp. 875–879.

[16] W. Fu, L. Song, and E. P. Xing, "Dynamic mixed membership blockmodel for evolving networks," in *International Conference on Machine Learning, ICML 2009, Montreal, Quebec, Canada, June*, 2009, pp. 329–336.

[17] S. Leonardi, A. Anagnostopoulos, J. cki, S. Lattanzi, and M. Mahdian, "Community detection on evolving graphs," 2016.

[18] H. H. Mao, C. J. Wu, E. E. Papalexakis, C. Faloutsos, K. C. Lee, and T. C. Kao, *MalSpot: Multi 2 Malicious Network Behavior Patterns Analysis*. Springer International Publishing, 2014.

[19] Y. R. Lin, Y. Chi, S. Zhu, H. Sundaram, and B. L. Tseng, "Facetnet: A framework for analyzing communities and their evolutions in dynamic networks," in , 2008, pp. 685–694.

[20] M. S. Kim and J. Han, "A particle-and-density based evolutionary clustering method for dynamic networks," *Proceedings of the Vldb Endowment*, vol. 2, no. 1, pp. 622–633, 2009.

[21] Y.-Y. Ahn, J. P. Bagrow, and S. Lehmann, "Link communities reveal multiscale complexity in networks," *Nature*, vol. 466, no. 7307, pp. 761–764, 2010.

[22] G. Palla, I. Derényi, I. Farkas, and T. Vicsek, "Uncovering the overlapping community structure of complex networks in nature and society," *Nature*, vol. 435, no. 7043, pp. 814–818, 2005.

[23] E. M. Airoldi, D. M. Blei, S. E. Fienberg, and E. P. Xing, "Mixed membership stochastic blockmodels," *Journal of Machine Learning Research*, vol. 9, no. Sep, pp. 1981–2014, 2008.

[24] M. Rosvall and C. T. Bergstrom, "Maps of random walks on complex networks reveal community structure," *Proceedings of the National Academy of Sciences*, vol. 105, no. 4, pp. 1118–1123, 2008.

[25] S. Gregory, "Fuzzy overlapping communities in net-

works," *Journal of Statistical Mechanics: Theory and Experiment*, vol. 2011, no. 02, p. P02017, 2011.

[26] S. Fortunato, "Community detection in graphs," *Physics reports*, vol. 486, no. 3, pp. 75–174, 2010.