Progression Analysis of Community Strengths in Dynamic Networks

Nan Du†, Jing Gao‡ and Aidong Zhang‡
†University at Buffalo (SUNY)
‡Computer Science and Engineering Department,
{nadu,jing, azhang}@buffalo.edu

Abstract—Community formation analysis of dynamic networks has been a hot topic in data mining which has attracted much attention. Recently, there are many studies which focus on discovering communities successively from each snapshot by considering both current and historical information. However, the detected communities are isolated at a certain snapshot, because these approaches ignore important historical or successive information. Different from previous studies which focus on community detection in dynamic networks, we define a new problem of tracking the progression of the community strength - a novel measure that reflects the community robustness and coherence throughout the entire observation period. The proposed community strength analysis provides significant insights into entity properties and relationships in a wide variety of applications. To tackle this problem, we propose a novel two-stage framework: we first identify communities via non-negative matrix factorization, and then calculate the strength of each detected community corresponding to each specific snapshot by solving an optimization problem. Experimental results show that the proposed approach is highly effective in discovering the progression of community strengths and detecting interesting communities.

I. INTRODUCTION

In recent years, there has been a growing interest in modeling and mining various kinds of dynamic networks such as biological networks, social networks, co-authorship networks and co-starring networks whose structures evolve over time. Among extensive work, community analysis in dynamic networks which focuses on detecting the communities successively from each snapshot by considering the historical information has recently attracted much attention [1]–[3]. However, all these detected communities are frozen and isolated at a specific snapshot. Thus we do not know when these communities were assembled or when they are going to disband. Aiming to answer these questions, we propose a novel measure called community strength.

In this paper, we claim that a community is with high strength if it has more internal interactions connecting the members of the community than the external interactions connecting to the rest of the world. Dense internal interactions and weak external interactions to the outside guarantee that the community is under a low risk of member change (current members leaving or new members joining). It is easy to understand that a strong community is somewhat strong if its members tie together closely and ignore the temptation from the outside world. On the contrary, we regard a friend community as a weak community, if it is likely to confront a member alteration situation. To illustrate this idea, Fig. 1(a) shows a toy example, where the nodes represented by the same geometric shape belong to the same community, solid lines represent internal interactions and dash lines represent external interactions. The circle community (i.e. nodes A, B, C and D) is considered to be stronger than the rectangle community (i.e. nodes E, F, G and H), because it involves less external attractions. Specifically, node H is having a close relationship with the diamond community (i.e. nodes I, J and K), which makes the rectangle community in the risk of losing its members. The higher strength score a community obtains, the less possible member alteration occurs in it.

Furthermore, community strength is a temporal measure whose value may change as the network evolves. For example, in Fig. 1(b) (i.e. the network in the 2nd snapshot) which evolves from Fig. 1(a) (i.e. the network at the 1st snapshot), the strength of the rectangle community decreases, since the internal connections become weaker and external connections become stronger. Let’s look at another real world example. Suppose a set of authors have collaborated closely from 2000 to 2006. During this period, they cooperated frequently among themselves and barely with others outside the community. However, after 2006, for the reason of interest changes, some authors’ attention has been attracted to some other fields. Thus the internal cooperation decreased and the external cooperation increased. In this case, this author community’s strength is high and stable during 2000-2006, but begins to decrease after 2006.

![Fig. 1: A Toy Example of Community Strength](image-url)

Discovering the progression of community strengths can offer significant insights in a variety of applications. It can help us discover some interesting community information which cannot be directly obtained from traditional community analysis. Now let us show two specific cases in detail.

**Strengths Progression in Actor Community:** As a strong actor community, the cooperation should be more frequent between the members themselves than between members and non-members. For example, considering the popular and long-running television sitcom ‘Friends’, its six main actors Jennifer...
Aniston, Courteney Cox, Matthew Perry, Matt LeBlanc, Lisa Kudrow and David Schwimmer collaborated closely when this sitcom was aired from 1994 to 2004. If we consider each year’s co-starring relationship as a snapshot, we can see that the strength of this community is very low before 1994 (little cooperation between them), and then dramatically increases and keeps stable from 1994 to 2004 (average 23 episodes each year). Finally, the strength of this community apparently becomes weaker after 2004 (much less cooperation comparing to previous years). Learning the strength progression of actor communities helps us better understand the actor cooperation history and entertainment industry.

**Strength Progression in Gene Community:** In the biological domain, the interactions between genes change gradually in dynamic gene co-expression networks. Thus the strength of gene communities also changes. For example, it has been reported that the expression profiling of some key genes is very likely to be a crucial trigger for the disease deterioration. Discovering the strengths of gene communities throughout a specific disease progression can help us find significant clues of disease progression analysis of gene communities and track the strength progression. This is the first work on analyzing the temporal community strength and related notations, and then formally define the community strength of a community.

**Problem Setting**

In this section, we first introduce the definition of community strength and related notations, and then formally define the problem. Before proceeding further, we introduce the notation that will be used in the following discussion: Let a matrix be represented with uppercase letter, e.g. $D$, $d_{ij}$ denotes the $ij$-th entry in $D$, $d_i$ and $d_j$ denote vectors of $i$-th row and $j$-th column of $D$, respectively. Now, let us start by introducing the definition of the community strength.

**Community Strength:** Given a network $G = (N,E,W)$, where $N$ is the set of nodes in this network, $E$ is the set of edges connecting the nodes, and $W$ is a symmetric weight matrix representing the weights on edges. Then, the community strength of a community $z$ can be defined as:

$$\text{Strength}(z) = \sum_{i \in N} \sum_{j \in N} w_{ij} \cdot \sum_{k \in z} \sum_{l \in z} w_{kl} - \left( \sum_{k \in z} \sum_{v \in N} w_{kv} \right)^2,$$

where $\sum_{i \in N} \sum_{j \in N} w_{ij}$ denotes the sum of all edge weights in the network, $\sum_{k \in z} \sum_{l \in z} w_{kl}$ denotes the sum of internal edge weights inside community $z$, and $\sum_{k \in z} \sum_{v \in N} w_{kv}$ denotes the sum of internal and external edge weights attached to nodes in community $z$. The rationale of Eq. 1 is that the strength of a specific community can be calculated by the differences between the sum of within community weights and the sum of outside community weights. Now, our problem can be defined as follows:

**Input:**

Temporal Networks: A series of undirected networks $G^t = (V^t, E^t, W^t)$ ($1 \leq t \leq T$), where each network has $N$ nodes (i.e. $|V^t| = N$ ($1 \leq t \leq T$)). For each specific snapshot $t$, $V^t$ is a set of nodes, $E^t$ is a set of interactions between these nodes and $W^t_{N \times N}$ is a symmetric weight matrix. For $v_i, v_j \in V^t$, $w_{ij}^t$ indicates the interaction frequency between nodes $v_i$ and $v_j$.

**Output:**

Community Pool Matrix: We summarize all the communities detected from each snapshot into an $N \times K$ community pool matrix $C$ where $K$ is the number of all the unique communities (i.e. $K = |C|$). In addition, $C$ equals to $C_1 \cup C_2 \cup ..., \cup C_T$ where $C_t$ ($1 \leq t \leq T$), which will be introduced, is the temporal community indicator matrix at snapshot $t$. Temporal Community Strength: Let a $K \times T$ matrix $A$ denote the temporal strength for all detected communities, where $a_{kt}$ refers to the strength of community $k$ at snapshot $t$.

From input to output, we need some nuisance parameters.

**Nuisance Parameters:**

Temporal Community Indicator Matrices: At a specific snapshot $t$, we denote the community indicator matrix $C^t$ ($1 \leq t \leq T$) as an $N \times K_t$ matrix where $K_t$ is the number of communities at snapshot $t$. If node $i$ is assigned to community $k$ at snapshot $t$, then $C_{ik}^t = 1$ and 0 otherwise. As we mentioned, all the temporal community indicator matrices $C_t^t$ ($1 \leq t \leq T$) compose the community pool matrix $C$. Note that the communities represented in this matrix can be either overlapping or non-overlapping.

Temporal Community Relationship Matrices: At a specific snapshot $t$, we denote the community relationship matrix $S_t$ as a $K_t \times K_t$ matrix. Note that $s_{ij}^t$ represents the similarity between community $i$ and community $j$ that are detected at snapshot $t$.

**Methodology**

In this section, we present our method for solving the problem of temporal community strength analysis. We begin by introducing the method of partitioning the network from each snapshot into communities in Section III-A, and show the methods of tracking the strength of each community at each snapshot and post processing work of our framework to handle some practical issues in the Section III-B.
A. Symmetric Non-Negative Matrix Factorization

Given a series of temporal networks \( G_t = (V_t, E_t, W_t) \) (1 \( \leq t \leq T \)), we first partition each network independently into multiple communities, and then collect all the detected communities from every snapshot together as a community pool. To detect communities at each temporal network, we use the Non-negative Matrix Factorization (NMF) [5], which is widely used in many applications such as pattern recognition, signal modeling, bioinformatics and text mining. There are two reasons to use NMF: first, it can be easily applied to both hard clustering (i.e. each object belongs to exactly one community) and soft clustering (i.e. each object can belong to multiple communities); second, it could uncover more precise underlying similarities between communities.

Given a network \( W_t \in \mathbb{R}^{N \times N} \), it can be decomposed into two components: \( C_t \in \mathbb{R}^{N \times K} \) and \( S_t \in \mathbb{R}^{K \times K} \) as:

\[
\min_{C_t \geq 0, S_t \geq 0} \left\| W_t - C_t S_t C_t^T \right\|^2, \quad \text{s.t.} \quad C_t C_t^T = I, \tag{2}
\]

where \( W_t \) is an \( N \times N \) symmetric matrix that demonstrates the interactions between objects at time \( t \), \( C_t \) is an \( N \times K_t \) community indicator matrix representing the probability of grouping an object into a community and \( S_t \) is a \( K_t \times K_t \) non-negative matrix providing the relationship between communities detected at time \( t \). The basic idea of minimizing Eq. 2 is that, at iteration \( t \) we first fix \( S_t \) and update \( C_t \) as:

\[
c_{t}^{ij} \leftarrow \frac{W_t C_t S_t C_t^T}{(C_t C_t^T W_t C_t S_t C_t^T)^{ij}}. \tag{3}
\]

Similarly, fixing \( C_t \), we can obtain the updated rule for \( S_t \) as:

\[
s_{t}^{ij} \leftarrow \frac{C_t W_t C_t^T}{(C_t C_t^T S_t C_t^T C_t S_t C_t^T)^{ij}}. \tag{4}
\]

We iteratively update \( C_t \) and \( S_t \) until convergence. It is worth noticing that the community indicator matrix \( C_t \) can be applied to both hard clustering and soft clustering.

Since it is hard to tell whether a community is strong or not without comparing with other communities, we put all the unique detected communities together into a community pool \( \tilde{C}_{N \times K} \). This community pool is used as a candidate set where all the communities are easily normalized and compared. Based on it, we intend to find out which communities are grouped closely and consistently and which communities are grouped temporarily. Note that, although we remove the duplicate communities which are exact the same in our case, other filtering methods could be used. For example, a threshold can be set that, if any pair of communities have a similarity over this threshold, we merge these two communities.

B. Temporal Community Strength Analysis

Now, we propose an integrated optimization framework that conducts community strength detection across snapshots. There are mainly two reasons why we use an integrated optimization framework rather than calculate the strength of each community individually at each snapshot. First, our framework is based on the smoothed assumption in which both current and historical network contribute to the community strength detection. Second, in the proposed framework, all communities’ strengths at a specific snapshot are brought into alignment, and then we can easily compare them. In our method, each community’s strength is weighted appropriately.

Based on Eq. 1, the strength of community \( z \) can be further reformulated in terms of the community pool matrix \( \tilde{C} \) step-by-step as follows:

\[
\text{Strength}(z) = \sum_{i,j=1}^{n} w_{ij} \sum_{i,j=1}^{n} w_{ij} \tilde{c}_{iz} \tilde{c}_{jz} - \left( \sum_{i,j=1}^{n} w_{ij} \tilde{c}_{iz} \right)^2 = \sum(W) \tilde{c}_{iz}^T W \tilde{c}_{iz} - \left( \sum_{i=1}^{n} d_{iz} \tilde{c}_{iz} \right)^2 \tag{5}
\]

where \( \sum(W) \) denotes the sum of weights for network \( W \) and \( W \) equals to \( \sum(W) + W \). Furthermore, \( D \) equals to \( d W \) where \( d \) is an \( N \times 1 \) vector such that each \( d_i \) is the sum of weighted degree of vertex \( i \). Employing Eq. 5, we formulate the task of discovering each detected community’s strength at a particular snapshot \( t \) as the following objective function:

\[
\min_{a_{zt}} J(a_{zt}) = \alpha \sum_{z=1}^{K} \log \left( \frac{1}{a_{zt}} \right) \tilde{c}_{iz}^T (\tilde{W}^t - D^t) \tilde{c}_{iz}^T + (1 - \alpha) \sum_{z=1}^{K} \log \left( \frac{1}{a_{zt}} \right) \tilde{c}_{iz}^T (\tilde{W}^{t-1} - D^{t-1}) \tilde{c}_{iz}^T + \gamma \sum_{z=1}^{K} a_{zt} - \mu_t, \tag{6}
\]

where \( \log \left( \frac{1}{a_{zt}} \right) \) determines the \( z \)-th community’s strength weight corresponding to the snapshot \( t \) and \( \mu_t \) is the estimated sum of community strengths with respect to the current snapshot. There are mainly two reasons for us using \( \log \left( \frac{1}{a_{zt}} \right) \) in the objective function. First, the logarithm function helps smoothing the strength values by limiting them within a small range. Second, it makes the objective function easier to solve. The stronger a community \( z \) at snapshot \( t \), the higher \( a_{zt} \) will be and hence \( \log \left( \frac{1}{a_{zt}} \right) \) will be lower. This means that higher weight will be associated with the community that is strong at the current snapshot.

Note that in Eq. 6, the first term measures the cost of all the detected communities in the community pool with respect to the current snapshot’s network, where a higher cost means worse strength of these communities. Similarly, the second term denotes the temporal smoothness in terms of the goodness of the current clustering result with respect to the previous network, where a higher temporal cost means worse temporal smoothness. In many real-world dynamic network applications, the networks are expected to change gradually and stably, such as geometric networks [6] and gene networks [7]. As a consequence, we expect a certain level of temporal smoothness between the community strengths in successive snapshots. On the one hand, we hope that the temporal community strength should depend on the current network. On the other hand, the temporal community strength should not deviate too dramatically from the previous snapshot’s network. Therefore, we adopt the temporal smoothness assumption in our framework. In Eq. 6, the overall cost of the objective
function is represented as the linear combination of the cost of community strength fitting to the current snapshot and the cost of community strength fitting to the previous snapshot. Thus $\alpha (0 \leq \alpha \leq 1)$ is a predefined parameter to reflect the users emphasis on the two snapshot cost. Usually, $\alpha$ could be assigned a relative large value when the networks are robust and evolve slowly, such as social networks. On the contrary, $\alpha$ should be assigned a relative small value when the target networks include noise and are likely to evolve swiftly, such as biological networks. The last term $\gamma \left(\sum_{z=1}^{K} a_{zt} - \mu_{t}\right)$ denotes the sum of community strengths of all detected communities at a specific snapshot $t$ is constrained by an estimated value $\mu_{t}$. For the sake of simplicity, we set $\mu_{t}$ as 1 in the experiments.

### PACS Algorithm Procedure

Now, we derive the solution for the community strength scores $a_{zt}$. Taking the partial derivative of Eq. 6 with respect to a particular $a_{zt}$ and setting the derivative to 0, we obtain Eq. 7 and Eq. 8.

$$a_{zt} = \frac{\alpha z t \left(\sum_{z=1}^{K} a_{zt} - \mu_{t}\right) \gamma}{\mu_{t}}$$

$$\gamma = \frac{\sum_{z=1}^{K} \left[\alpha z i t \left(\sum_{z=1}^{K} a_{zt} - \mu_{t}\right) \gamma\right]}{\mu_{t}}$$

Finally, the solution for $a_{zt}$ is shown as Eq. 9. In this equation, the numerator measures the strength of the community $z$ at the snapshot $t$ and integrates both the current and historical information. The denominator represents the overall strength across all the communities at snapshot $t$, which serves as a normalization factor. Furthermore, $\mu_{t}$, as mentioned before, controls the overall community strength at a specific snapshot. The intuition behind Eq. 9 is that the communities which have firm structures at the current snapshot will be assigned a large strength score; while the community whose structures are loose will receive a low value. The proposed algorithm is shown in Algorithm 1, and we name our method as PACS (Progression Analysis of Community Strength).

$$a_{zt} = \frac{\sum_{z}^{K} \left[\alpha z i t \left(\sum_{z=1}^{K} a_{zt} - \mu_{t}\right) \gamma\right]}{\mu_{t}}$$

### Top-K strongest/weakest communities

Based on the proposed Algorithm 1, we can answer the question ‘How does each community’s strength evolve over the observation period?’. In addition, our method can be easily extended to find the top-K strongest communities, which can answer the questions ‘What are the top-K strongest or weakest communities during the observation period?’.

Through Algorithm 1, we obtain the community strength for each detected community at each snapshot. Based on it, we can get an overall community strength, which is useful to identify interesting communities that are the strongest/weakest throughout the entire observation period. There are mainly two methods to aggregate the temporal community strength scores: unweighted and weighted. It is easy to understand that, in an unweighted case we just aggregate each temporal score equally, i.e. $\sum_{t=1}^{T} a_{zt}$. However, in some special cases, the community strength is more important at some particular snapshots, e.g. the early stage of cancer. In such a case, we should give different weights to different snapshots and the aggregation function can be defined as $\sum_{t=1}^{T} h_t a_{zt}$, where the $h_t$ is the weight for the specific snapshot $t$. In addition, when choosing the top strongest or weakest communities we may also want to consider the size of the communities. Because when the target networks are very sparse, the penalty from the external edges may be very small, the results are thus biased to the communities with large size. Accordingly, the aggregated function for community $z$ can be redefined as: $\frac{\sum_{t=1}^{T} a_{zt}}{|C_z|}$ or $\sum_{t=1}^{T} h_t a_{zt}$.

### IV. Experiments

In this section, we report experimental studies based on both synthetic and real-world datasets. We evaluate our method by comparing our detected the strongest/weakest communities and community strength ranking with the ground truth. Moreover, we perform comprehensive analysis to justify the top-K strongest communities returned by the proposed algorithm.

#### A. Synthetic Dataset

We start with a synthetic dataset, which is generated according to the method mentioned in [3]. At the $i^{th}$ snapshot, we generate 100 nodes, which are divided into five communities of 20 nodes each. We generate data for total 30 consecutive snapshots. From the $2^{nd}$ to the $30^{th}$ snapshots, edges are added randomly with a higher probability $p_{in}$ for within-community edges and a lower probability probability $p_{out}$ for between-community edges. In this study, we set a two-tuple parameter $(p_{in}, p_{out})$ for these five communities as $C_1 = (0.22, 0.05)$, $C_2 = (0.2, 0.07)$, $C_3 = (0.18, 0.09)$, $C_4 = (0.16, 0.11)$ and $C_5 = (0.14, 0.13)$.

#### Baselines

Since there is no previous method that can be directly used to handle the same problem, we compare the proposed algorithm with the variations of itself and the variations of previous relevant method.

- **PACS without**: The first baseline, which we call **PACS without**, uses the proposed method but does not consider the smoothness assumption. Comparison with
will demonstrate the importance of the smoothness assumption.

KNN: The second baseline is the K-nearest neighbors (KNN). At each snapshot, after we use a KNN to detect the communities from it, we also calculate the strengths for them. Then, strengths of any communities can be calculated via the linear combination of the top-K most similar communities’ strengths. In KNN, the formula for strength of community \( i \) at time \( t \) can be represented as \( \sum_{j=1}^{K} h_{ij} a_{jt} \). Note that \( h_{ij} \) here is the similarity between two communities calculated with Jaccard coefficient and \( \sum_{j=1}^{K} h_{ij} = 1 \). Comparison with KNN will help us understand whether a method based on the local view can detect the strong communities.

\( \text{CID:} \) The third baseline is based on community internal density, which we call \( \text{CID} \) for short. As defined in [8],
\[
\text{CID}_c^i = \frac{\sum_{j \in c} \sum_{j \in c^i} w_{ij}}{|c||c^i| - 1}/2
\]
measures the internal edge density of the cluster \( c \). Since \( \text{CID} \) is also proposed to measure the community quality, comparing with \( \text{CID} \) can help us understand which community quality index can better measure the community strength.

Evaluation & Performance on Synthetic Dataset

Due to the way we generated the synthetic data, we have known community \( C_1 \) has the largest gap between \( p_{in} \) and \( p_{out} \), which makes it the strongest community throughout all the snapshots. Thus, we can directly compare the strongest community discovered by each method with \( C_1 \). The higher similarity the detected strong community with \( C_1 \) denotes the corresponding method is more precise. In this experiment, we use Jaccard coefficient to measure the community similarity. On the contrary, since all these five communities in the synthetic data are more or less well separated, the weakest community is the one composed with the members coming uniformly from these five communities. To measure the distribution of members, we use the entropy, the higher the entropy is, the weaker the discovered community is.

Table I shows both the Jaccard coefficient for the strongest community and the entropy for the weakest community for the four algorithms. From the table we can see that the proposed method clearly outperforms the baselines at both strong and weak community detection. Furthermore, from the results, we can see that the smoothness assumption contributes to the performance improvement.

<table>
<thead>
<tr>
<th>Method</th>
<th>Jaccard coefficient (Strong Community)</th>
<th>Entropy (Weak Community)</th>
</tr>
</thead>
<tbody>
<tr>
<td>PACS</td>
<td>0.95</td>
<td>1.38</td>
</tr>
<tr>
<td>PACS without</td>
<td>0.90</td>
<td>1.33</td>
</tr>
<tr>
<td>KNN</td>
<td>0.85</td>
<td>1.1</td>
</tr>
<tr>
<td>CID</td>
<td>0.85</td>
<td>1.33</td>
</tr>
</tbody>
</table>

B. Social Evolution Dataset

The social evolution dataset was collected by MIT human dynamics lab [9], which closely tracked the daily life of 80 students in a dormitory with mobile phones. We construct the student interaction networks by using the interaction information. In these networks, nodes are students and an edge exists between them if the corresponding students have the interaction through the following three ways: call, message or music sharing in a given period of time. The weight of each edge is the number of interactions. It is easy to see that, the more frequently two students interact with each other, the higher weight is assigned on the edge connecting them. In addition, there are five snapshots which correspond to the time before 10/19/2008 - 10/19/2008 (T1), 10/20/2008 - 12/13/2008 (T2), 12/14/2008 - 3/5/2009 (T3), 3/6/2009 - 4/17/2009 (T4) and 4/18/2009 - 5/22/2009 (T5).

Note that the number of communities at a specific time \( t \) is automatically determined by the modularity function [10]. To determine the best community number \( K \) at time \( t \), we tried different candidates for \( K \) in a specific range and used the best one. From these student interaction networks, we hope to rank these friend communities based on their strengths. In other words, we try to find out the strong friend clans from their regular social interactions.

Evaluation of temporal community strength is difficult due to lack of ground truth. Fortunately, besides the student interaction information, this dataset also provides a series of temporal surveys about the friendship between students, which can be used to validate our discoveries. The surveys were mainly made on five dates: 10/19/2008, 12/13/2008, 3/5/2009, 4/17/2009 and 5/22/2009, that is the reason why we cut the snapshots of student interaction networks in the way mentioned above. In each survey, every student needs to indicate his/her current relationship level with others in six kinds of surveyed relationships, which are sorted and weighted by us in view of the intimacy degree.

Evaluation & Performance on Social Evolution Dataset

We first calculate the gap between the average inside-community friendship and the average outside-community friendship for each community in the community pool \( C \), and then we sort these communities based on their gaps as a ranking list \( L_1 \). Note that the higher a community is ranked, the stronger it is. It is easy to understand that a strong friend clan should simultaneously obtain the close relationships among the members and obtain a relatively weaker relationship with non-members. This dense and isolated situation ensures a low probability of current members leaving and new members joining, which makes it a strong friend clan. Also, we calculate the community strengths for each community in the community pool \( C \) with the proposed method and baselines, and then also rank them as a ranking list \( L_2 \). Since \( L_1 \) denotes the ground truth of the communities’ strengths ranking, we can directly compare the predicted result \( L_2 \) with \( L_1 \).

Two approaches have been used to measure the similarity between ranked list \( L_1 \) and \( L_2 \): a global one and a local one. For the global one, we want to see how close the overall similarity is between \( L_1 \) and \( L_2 \). To measure it, we use the rank correlation proposed in [11], which is also commonly referred as Kendall’s tau (\( \tau \)) coefficient. The Kendall’s tau coefficient is in the range of [-1,1], where two identical rankings has value 1, and the opposite rankings (i.e. when one ranking is the reverse of the other) has value -1. Besides measuring the proposed method with a global view, we are also interested in the method’s local precision at the two extremes (strongest and weakest) in a ranking list. To measure the accuracy of detected communities in the top/bottom \( x \) elements, we use a cover rate function of
\[
\frac{|L_1^x \cap L_2^x| + |L_2^x \cap L_2^x|}{|L_1^x| + |L_2^x|},
\]
where \( L_1^x \) denotes the...
elements of the top $x$ ratio of list $L$ and $L^{B_x}$ denotes the elements of the bottom $x$ ratio of list $L$.

To measure the influence from the different ratio $x$, we vary the mutation ratio $x$ as 10%, 15%, 20%, 25%, and 30%. The experimental results of global and local ranking measurement comparing with baselines mentioned in Section IV-A are shown in Fig. 2(a) and Fig. 2(b), respectively. It can be observed from these figures that the proposed method outperforms other baselines consistently in both the global measure and local measure.

To fully evaluate how the temporal smoothness parameter $\alpha$ affects the performance, we increase $\alpha$ from 0 to 1 with a step of 0.1 and report the Kendall’s tau value of the detected community ranked list in Fig. 2(c). We can observe that, as $\alpha$ increases, which means we emphasize more on the current network, we get a hill-shape curve, which demonstrates that our framework is able to optimize the results based on the temporal smoothness assumption.

V. RELATED WORK

Community analysis of dynamic networks has been extensively studied in various research areas. However, most existing community research on dynamic networks focuses on several limited topics, such as community discovery and community evolution pattern detection for the objects. Chi et al. [2] and Lin et al. Gupta et al. [1] investigated and tackled the problem of identifying evolutionary community outliers given the discovered communities from two snapshots of an evolving dataset. Their problem is to find out the node-community pairs with a high outlierness score from two snapshots, whose focus is different from ours. In addition, Newman and Girvan [10] proposed the modularity function, which measures the quality of partitioning a graph into multiple communities. In particular, a graph is considered with high modularity if it has dense connections between the nodes within the communities but sparse connections between nodes across different communities. The way how they measure the graph is similar to our community strength’s definition. However, this modularity function is only used as a global index which measures the quality of a particular clustering of nodes in a graph or the standard to partition the objects into communities [12].

VI. CONCLUSIONS

In this paper, we introduced a new problem of analyzing the progression of community strengths. Community strength is a temporal measure which represents the probability that a particular community has a firm structure at the current snapshot. We proposed a two-stage framework which includes community detection and community strength analysis. This method can provide reliable and consistent community strength scores which are not only less sensitive to short-term noises in the current network but also adaptive to long-term networks evolution by considering the temporal smoothness. Moreover, the results of community strength analysis can help us find the top-K strongest or weakest communities throughout entire observation. Experiments on both synthetic and real dynamic datasets demonstrated the superior performance of the proposed method compared with other baseline methods in finding the top-K strongest or weakest communities.

REFERENCES


