# GRAPH SUBSPACE TRACKING FOR ONLINE COMMUNITY CHANGE DETECTION

*Minghe Zhang*[⋆] *Yao Xie*[†]

[⋆] School of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, GA, USA.
[†]School of Industrial and Systems Engineering, Georgia Institute of Technology, Atlanta, GA, USA.

## ABSTRACT

In this paper, we present a novel subspace tracking approach for graphs on Grassmann manifolds. We consider a scenario where a graph might change some of its connections in a time period. We show that an efficient gradient descent method on a Grassmann manifold can be used to implement online community detection by updating the subspace representation each time a new graph laplacian matrix is considered. The dynamic graph is simulated first by randomly generating a series of Stochastic Block Models (SBM) with some predetermined communities. Our generic framework then further extends to solve change point detection and anomaly detection problems. We adopt several change detection procedures and discuss the trade offs between them.

***Index Terms*—** Grassmann manifold, subspace tracking, community detection, gradient descent, change detection

## 1. INTRODUCTION

Community detection problem has been a popular topic in recent years with the development of machine learning and artificial intelligence. It also has broad applications in the area such as language processing, bio-information retrieval and social network analysis. Traditionally, community detection is applied to a single graph and gives back the community ID for each node. [1] presents a comprehensive introduction for single graph's community detection tasks.

However, sometimes a static graph can only reveal parts of the structural features through community detection, so some approaches that merge multiple graphs or gradually learn from these graphs are developed. [2] implements clustering on multi-layer graphs by merging all the individual graphs into one representative graph. Simensen's work [3] for online community detection is based on the adjacency matrix of the dynamic graphs. [4] reconstructs the planted clustering given the ability to query for small pieces of local information about the graph, at a limited rate.

Our paper has improved efficiency in community detection as it considers one single graph at one time and map it into a subspace, then updates the optimized function using gradient descent on the grassmann manifold. Furthermore, our approach can be applied to online community change detection problem and has been proved to have a low error rate and short detection delay time. Our probabilistic model to simulate the change of the community structure of a graph is based on the stochastic block model (SBM) ([5], [6]). It is a widely accepted model of probabilistic networks for the study of community-detection methods, which generates graphs with an embodied community structure.

## 2. PROBLEM STATEMENT

Assume a graph with fixed N nodes and an observed sequence of independent adjacency matrices over time $W_1$, $W_2$, $W_3$,... $W_t$ where $W_t \in \mathbb{R}^{n\times n}$ is the adjacency matrix of this dynamic graph at time $t$. We refer to changes of time as the emergence or disappearance of edges or the significant changes of the edges' weights in most cases. But no matter how they changes, the establishment of edges forms a similar rule. For instance, the nodes from the same potential community in the graph are connected with a higher probability $p_{int}$ while the nodes from different community could form an edge with a lower probability $p_{ext}$ .

The structure community of this graph may change from time to time. A large community may split into several small ones which also have a chance to combine together. Nodes from one community may get into another on after some certain change point.

Our goal is to 1) find out these potential communities by learning from the graphs; 2) detect the change of the community structure as quickly as possible.

## 3. SEQUENTIAL CHANGE DETECTION

In this section, we propose an algorithm which firstly uses spectural clsutering on grassmann manifold to extract the useful structural information of the graph's community in lower dimention. Then based on the statistic achieved from that, we implment several change detection algorithms and compare the accuracy between them. The proposed method's scheme is shown in Fig. 1 .

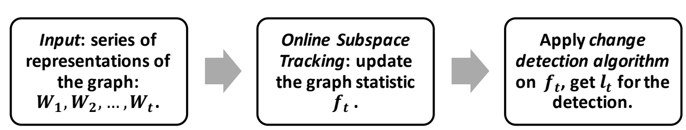



**Fig. 1**. Scheme of the Online Commuity Change Detection

### 3.1. Extract Statistic from Seqence of Graph Data

Since each of the graph data in a sequence is an adjacency matrix $W_t \in \mathbb{R}^{n\times n}$, it would be quite time consuming when it comes to online change detection. As a result, we compress them into a martrix $Q \in \mathbb{R}^{n\times k}$ in which $k$ is usually much smaller than $n$ while maintaining most part of the information from these $W_t$. The way to achieve this is by spectral clustering which is discussed in detail in [7].

However, if we just apply spectural clustering on each graph, we will get a seqence of $Q_t$, which is also time consuming so that we perform an update procedure each time on $Q \in \mathbb{R}^{n\times k}$ on a grassmann manifold. The Grassmannian denoted as $Gr(k, V)$ is a space which contains all $k$-dimensional linear subspaces of the $n$-dimensional vector space $V$. As a compact Riemannian manifold, its geodesics can be computed as indicted in [8]. Our target matrix $Q$ can then be represented as a point in the grassmann manifold. Each time a new adjacency matrix $W_t$ of a graph at time comes in, we compute graph laplacian of it as $L_t = D_t^{-\frac{1}{2}}(D_t - W_t)D_t^{-\frac{1}{2}}$ where $D_t$ is the degree matrix of the graph at time $t$. The optimization problem becomes finding optimized $Q$ such that:

$$\min_{Q\in\mathbb{R}^{n\times k}} \sum_t \mathrm{tr}(Q^T L_t Q), \quad s.t. \quad Q^T Q = I \tag{1}$$

We consider the problem for every time slot and let function

$$f(Q) = \mathrm{tr}(Q^T L_t Q) \tag{2}$$

And the derivative of $f$ with respect to the elements of $Q$ according to [9] is

$$\frac{df}{dQ} = \frac{d(\mathrm{tr}(Q^T L_t Q))}{dU} = (L_t + L_t^T)Q. \tag{3}$$

Then we Use Equation (2.70) in [8] to get the gradient of function $f(Q)$ on grassmann manifold from Eq. 3:

$$\nabla f = (I - QQ^T)\frac{df}{dQ} = (I - QQ^T)(L_t + L_t^T)Q. \tag{4}$$

Gradient descent algorithm along a grassmann manifold is given by Equation (2.65) in [8], proving that it is a function of the singular values and vectors of $-\nabla f$, so suppose we have get the reduced Singular Value Decomposition (SVD) of $-\nabla f = U\Sigma V^T$, we can write the updating function with a step size $\eta$ as:

$$Q(\eta) = (QV\ U)\begin{pmatrix}\cos\ \Sigma\eta \\ \sin\ \Sigma\eta\end{pmatrix} V^T \tag{5}$$

Here we update $Q$ with a step size $\eta$ to get closer to the local minimum on the grassmann manifold. The complete algorithm is shown in Algorithm 1.

**Algorithm 1** Online Learning for Dynamic Graphs

**Input:** Weighted adjacency matrix at time $t$ denoted as $W_t$, the total number of iterations $T$, number of communities $k$, A set of step sizes $\eta_t$.

**Output:** Subspace Representation $Q$ of the graph.

```
Initialize Q randomly.
for t = 1, ..., T do
    Compute current graph's laplacian matrix L_t.
    Compute -∇f = -(I - QQ^T)(L_t + L_t^T)Q.
    Compute SVD of -∇f = UΣV^T.
    Get an Update of Q each time by using Eq.5.
return Q
```

**Choice of Step Size:** For the problem of gradient descent of community detection, constant step size and decreasing step size can both be considered as efficient tools. Constant step is slower at the beginning but it is more stable to detect changes of community structures. In our numberical experiments, we take constant step size $s = 0.01$.

### 3.2. Community change detection

In this section, we provide two methods for detecting the change of the graph community structure which will cause a great change to the means of the observation. Here the observation we use are the optimal objective function in the section 3.1 which is $f_t = tr(Q^T L_t Q)$. We find that if a graph's structure changes such like it is split into two separate community or the community size is changing dramatically, the mean of $f_t$ would be increase or decrease until the optimization procedure converges again. Such a gradual change can be well approximated by a slope change[10] or a direct shift[11] in the means of function $f_t$ in Eq. 2.

*Method 1: Slope change detection based on GLR procedure.* First, we formulate the problem as a hypothesis testing problem:

$$\begin{aligned} H_0: &\quad f_t \sim \mathcal{N}(\mu, \sigma^2),\ t = 1, 2, ..., T, \\ H_1: &\quad f_t \sim \mathcal{N}(\mu, \sigma^2),\ t = 1, 2, ..., \kappa, \\ &f_t \sim \mathcal{N}(\mu + c(t-\kappa), \sigma^2),\ t = \kappa+1, \kappa+2, ..., T, \end{aligned} \tag{6}$$

Here $f_t$ is the our observed statistics. We assume that before the change of the graph structure, $f_t$ has a known mean $\mu$ and a known variance $\sigma^2$. And then it changes linearly from the change point time $\kappa + 1$. We want to stop as soon as possible after a change occurs and avoid false alarm when no change happens. According to Equation (7) from [10], we gives the Generalized likelihood ratio (GLR) for the observations:

$$\ell(k, t) = \frac{\left(\sum_{i=k+1}^{t} (i-k)(f_i - \mu)/\sigma\right)^2}{2\sum_{i=1}^{t-k} i^2} \tag{7}$$

Then we define the $GLR$ procedure as:

$$X(t) = \max_{\max\{0,t-w\}\leq k<t} \ell(k,t) \tag{8}$$

Here $w$ is the sliding window size. Then we just need to set a reasonable threshold value so that each time $X(t)$ is above this value, an alarm is raised indicating the happening of a change. Its validation is proved in the following section with an simulation using SBM graphs.

*Method 2: Multiple change-point detection.* Here we implement the approach of [12] on our problem. Since it is a detection problem rather than a segmentation problem. We track a continuous statistic of the objective variable and then set the threshold to detect the possible change of communities.

Assume the mean of the observed variable is a step function with constant values on each stable period. In our problem, we continue use trace $f_t$ as the observed variable. Let $S_f = \sum_1^f f_f$, and we want to find $i, j, k$ such that the trace's mean is different in interval $[i, j]$ and $[j, k]$. According to Eq.(1.1) in [12]:

$$Z(i,j,k) = \frac{S_j - S_i - (j-i)(S_k - S_i)/(k-i)}{[(j-i)(1-(j-i)/(k-i))]^{1/2}} \tag{9}$$

We extract statistic $\ell(t)$ from $Z(i, j, k)$ following the detection procedure in Algorithm 2:

**Algorithm 2** Multi-Segmentation algorithm based on Meanshift change

**Input:** Pre-calculated $Z(i, j, k)$, threshold $th$, the total number of iterations $T$
**Output:** $\ell \in \mathbb{R}^T$

```
set i = 1, j = 2, k = 3, ℓ = zeros(T).
while k < T do
    Find j_opt between i and k that maximizes Z(i, j, k).
    ℓ(k) = Z(i, j_opt, k).
    if Z(i, j_opt, k) ≥ th, then i = k.
    k = k + 1.
return ℓ
```

## 4. EXPERIMENTAL RESULTS

In this section, we use artificial graph model called stochastic block model (SBM) to test our algorithm. We show that both of our community detection and change detection algorithms reveal high efficiency and accuracy.

*Stochastic Block Model (SBM) simulation.* The stochastic block model (SBM) is a random graph model with planted clusters. Here we use this model to finish the community detection as well as change point detection tasks with our proposed algorithms in this paper.

*SBM for online community detection.* For community detection problem, we model the graph to contain 3 communities and each one has 100 nodes in it. We set the step size to be constant as $\eta = 0.01$ and the nodes inside the same community are connected with probability $p_{int} = 0.5$ and the nodes from different community form an edge with a probability $p_{ext} = 0.3$. The initial target subspace representation of the graph $Q$ is generate randomly with all its entries between 0 and 1. Fig.2 shows the error rate of the community detection

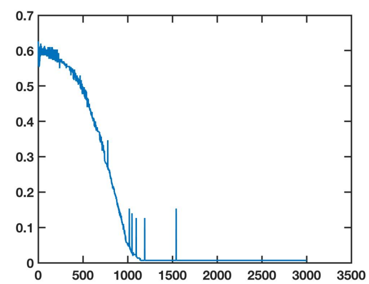



**Fig. 2**. Convergence Process of the Community Detection's Error Rate using our Online Learning Method.

task. The error rate is defined as the ratio of the number of nodes which lies in a wrong community to the node number in the whole graph.

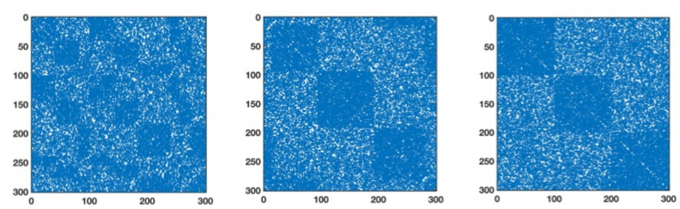
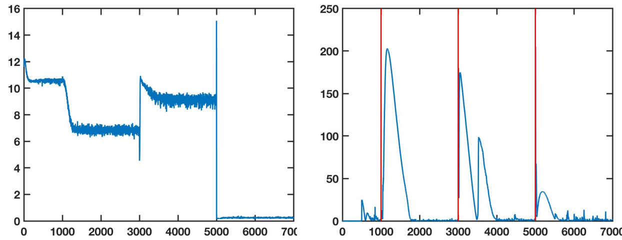


**Fig. 3**. Re-ordered adjacency matrix representation based on community detection results at different time t = 200, 1000, 2000.

Fig.3 is the representation of the re-ordered adjacency matrix. We put the nodes from the same community at each time $t$ to be the neighbors in its adjacency matrix representation. From above we can see after 1000 iterations the base form of three communities can be seen blearily and at 2000 iteration the community structure is shown quite clearly.

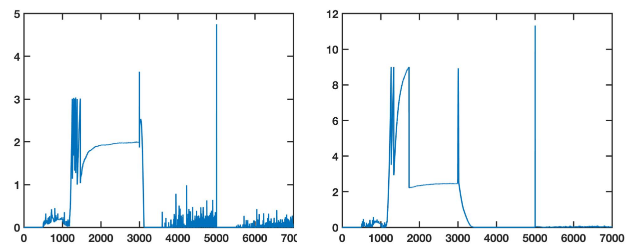
*SBM for change point detection.* In this experiment, we design a graph which changes its community structure for two times. The graph consists 100 nodes. We set $p_{ext} = 0.2$ and $p_{int} = 0.8$ and there are 6000 time slots in total. There are three community changes in total. In the very beginning, all the nodes are from one single community. Then at $t = 1000$ the community begins to split into two equally sized communities where each has 50 nodes. And at $t = 3000$ one of the community begins to contain 90 nodes while the other one contains only 10 nodes. Finally at $t = 5000$ the two communities are merged into one single community.

Fig.4 shows the results of slope change detection and Fig.5 is the results of multi-segmentation based detection. Although the multi-segmentation based detection is faster, the slope change detection gives lower detection delay and lower false alarm rate. So there's a trade off on how to choose the algorithm.



(a) Optimized Function F (b) GLR procedure X

**Fig. 4**. Change detection result for a graph with changed communities using slope change detection approach.



(a)Threshold equals 3 (b)Threshold equals 9

**Fig. 5**. Multi-segmented detection experiment with different choices of the threshold value.

## 5. NUMERICAL EXAMPLES

In this section, we provide a theoretical approximation to determine the choice of threshold $b$ of our proposed methods based on the fixed Average Run Length (ARL) and then gives the relation between Expected Detection Delay (EDD) and the probability of the connection between edges in SBM model.

### 5.1. ARL based threshold selection

Now we want to make an estimation of the ARL of our Slope change detection. Denote the ARL as the expectation of average run length when there is no change. $ARL = E(T) = \lambda$. It follows the polynomial distribution so that we can get $T \sim exp(1/\lambda)$, then we define $\hat{p}$ to be the average false alarm probability in a $length = 500$ period with threshold $b$.

$$\hat{p} = P(T \leq 500) = 1 - e^{-500/\lambda}$$

So use the formula above, we are able to get an approximated $\hat{p} = 0.0952$ based on a fixed $ARL = 5000$. Repeat the experiment for 1000 times then we can get it expected $b = 22.5$ that the ARL is as expected.

### 5.2. Simulation of the EDD

We define $\delta_p = p_{int} - p_{ext}$ to represent the strength and compactness of a community. From the above section we get threshold value $b$ so that its ARL is approximately 5000. Here we fix $b$ and explore the relationship between EDD and $\delta_p$. We repeat the SBM experiment for 1000 times and let the $\delta_p$ ranging from 0.3 to 0.8. Then we get the results in Fig 6. It can be seen that EDD reduces roughly linear with the increase of $\delta_p$ which indicates the EDD approximately linearly depends the on strength of the community structure.

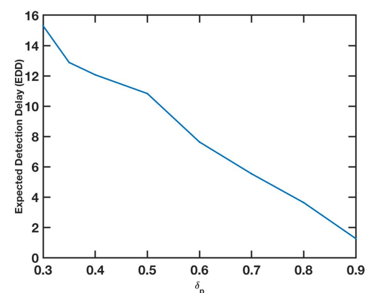
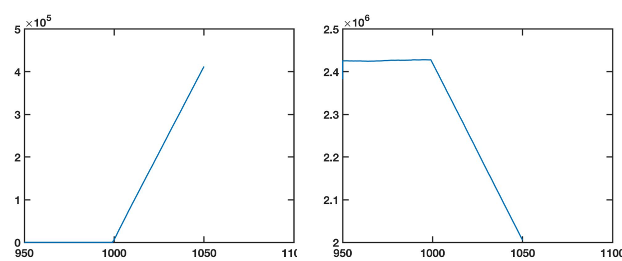



**Fig. 6**. Experimental Approximation of EDD based with respect to $\delta_p$.

To prove the efficiency of the algorithm, we compare our work with [3] and find our work has better scalability and higher efficiency. The earlier work can only detect increase of the number of communities as is the process of merging communities but it cannot detect a reduction in the number of communities (shown in Fig.7). our approach can detect both the increase and decrease of communities with a impulse signal above the predefined threshold. And our average speed for each iteration is **25** times faster.



(a) Communities are merged (b) Community is splited

**Fig. 7**. Simenson's online community detection approach[3]'s result. The number of nodes in the graph is 100 and the left figure is based on the procedure that two communities are merged into one at time 1000. The right one indicates the reverse procedure which one whole community is splited into two separate parts.

## 6. REFERENCES


[1] Santo Fortunato, "Community detection in graphs," *Physics reports*, vol. 486, no. 3-5, pp. 75–174, 2010.

[2] Xiaowen Dong, Pascal Frossard, Pierre Vandergheynst, and Nikolai Nefedov, "Clustering on multi-layer graphs via subspace analysis on grassmann manifolds," *IEEE Transactions on signal processing*, vol. 62, no. 4, pp. 905–918, 2014.

[3] David Marangoni-Simonsen and Yao Xie, "Sequential changepoint approach for online community detection.," *IEEE Signal Process. Lett.*, vol. 22, no. 8, pp. 1035–1039, 2015.

[4] Aris Anagnostopoulos, Jakub Lacki, Silvio Lattanzi, Stefano Leonardi, and Mohammad Mahdian, "Community detection on evolving graphs," in *Advances in Neural Information Processing Systems*, 2016, pp. 3522–3530.

[5] Emmanuel Abbe, Afonso S Bandeira, and Georgina Hall, "Exact recovery in the stochastic block model," *arXiv preprint arXiv:1405.3267*, 2014.

[6] Emmanuel Abbe, "Community detection and stochastic block models: recent developments," *arXiv preprint arXiv:1703.10146*, 2017.

[7] Ulrike Von Luxburg, "A tutorial on spectral clustering," *Statistics and computing*, vol. 17, no. 4, pp. 395–416, 2007.

[8] Alan Edelman, Tomás A Arias, and Steven T Smith, "The geometry of algorithms with orthogonality constraints," *SIAM journal on Matrix Analysis and Applications*, vol. 20, no. 2, pp. 303–353, 1998.

[9] Kaare Brandt Petersen, Michael Syskind Pedersen, et al., "The matrix cookbook," *Technical University of Denmark*, vol. 7, no. 15, pp. 510, 2008.

[10] Yang Cao, Yao Xie, and Nagi Gebraeel, "Multi-sensor slope change detection," *Annals of Operations Research*, vol. 263, no. 1-2, pp. 163–189, 2018.

[11] Yao Xie and David Siegmund, "Sequential multi-sensor change-point detection," in *Information Theory and Applications Workshop (ITA), 2013*. IEEE, 2013, pp. 1–20.

[12] Xiao Fang, Jian Li, and David Siegmund, "Segmentation and estimation of change-point models," *arXiv preprint arXiv:1608.03032*, 2016.