Anomaly Detection in Dynamic Graphs

guest lecture by Shenyang(Andy) Huang

October 14th 2022

Outline

1. **introduction to anomaly detection in graphs**
2. anomaly detection in dynamic graphs
3. laplacian change point detection for dynamic graphs
4. multi-view change point detection for dynamic graphs
5. Fast and Attributed Change Detection on Dynamic Graphs with Density of States

Definition of an Anomaly

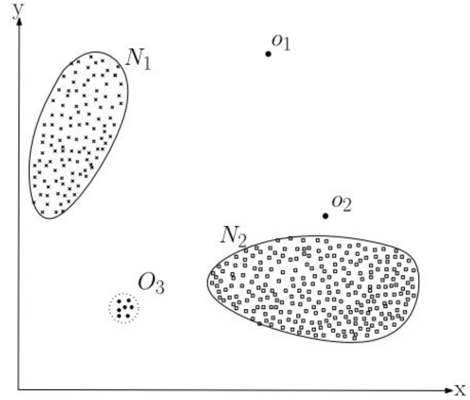
An anomaly is “**an observation** that differs so much from other observations as to arouse suspicion that it was generated by a **different mechanism.”**

said Douglas M Hawkins in 1980.

reference:
Douglas M Hawkins.Identification of outliers. Vol. 11. Springer, 1980

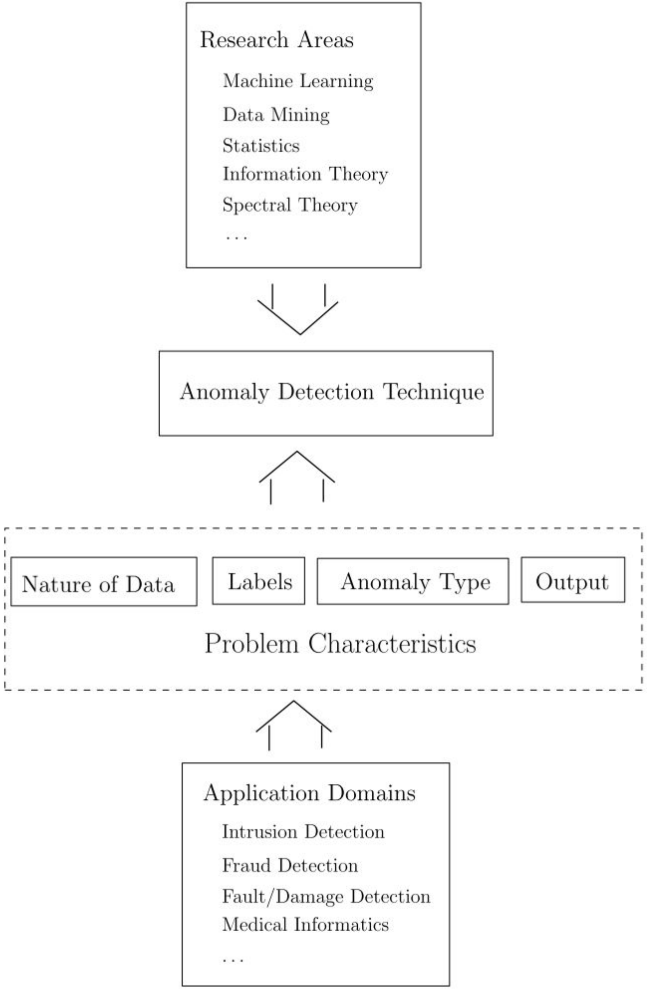
In practice, concrete definition can depend on:

1. The task of interest
2. The nature of the network



A simple example of anomalies in a 2-dimensional data set.

Reference: Anomaly Detection: A Survey



Key components associated with anomaly detection

Reference: Anomaly Detection: A Survey

Data Types

Categorized by relationships between data points

1. Point data
    a. No relations between points
2. Sequential data
    a. Linearly ordered
3. Spatial data
    a. Ordered by spatial location
4. **Graph data**

Anomalies in Graph

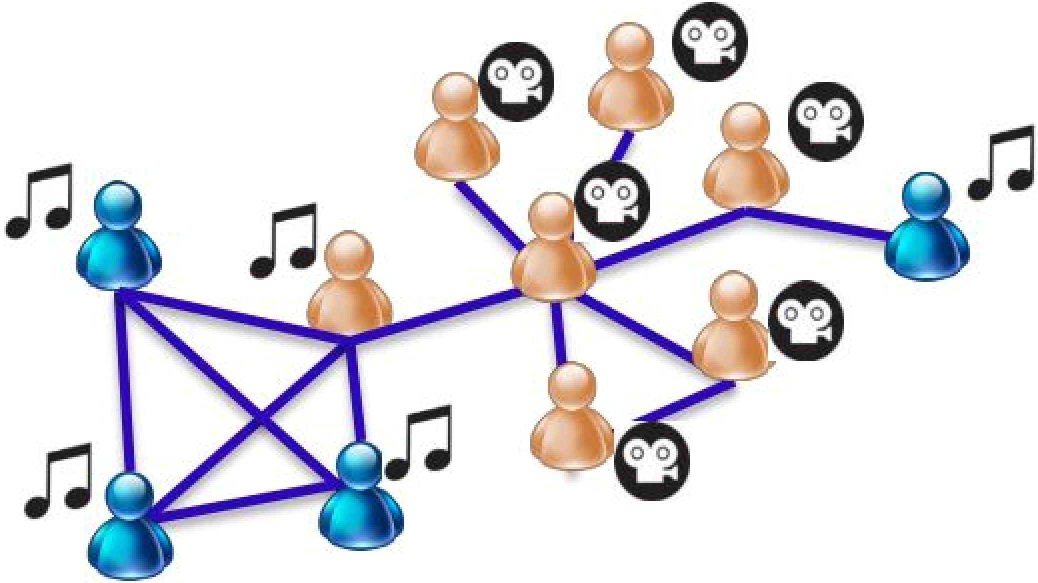


Figure from: :
Akoglu L, Tong H, Koutra D. Graph based anomaly detection and description: a survey. Data mining and knowledge discovery. 2015 May 1;29(3):626-88

Anomalies in Graph

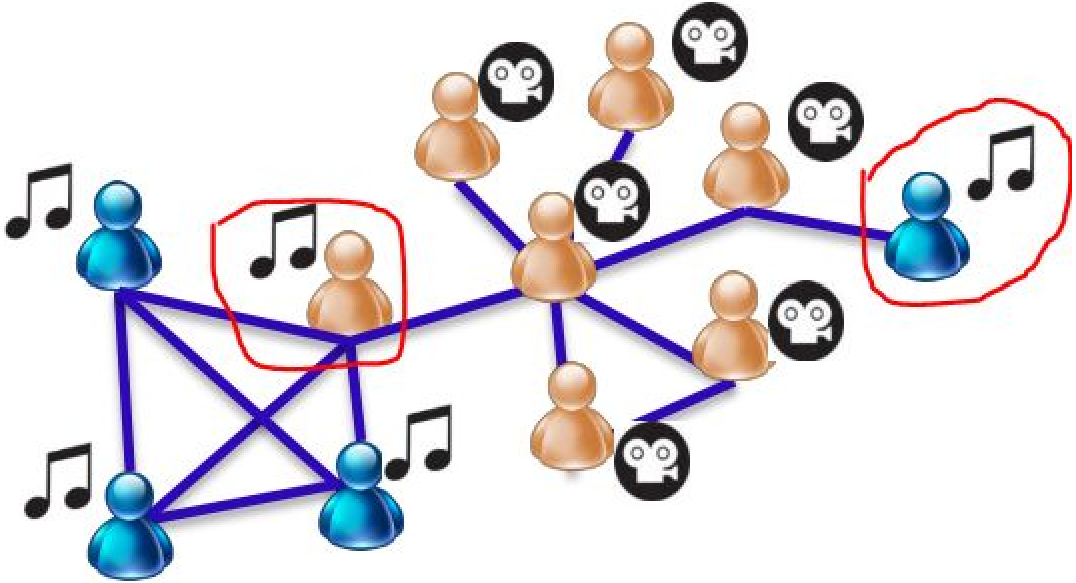


Figure from: :
Akoglu L, Tong H, Koutra D. Graph based anomaly detection and description: a survey. Data mining and knowledge discovery. 2015 May 1;29(3):626-88

Static Graph Anomalies

Definition:

Given a graph $\mathbf{G} = (\mathbf{V},\mathbf{E})$

Find the **nodes / edges / subgraphs** which are **rare and different** or deviate significantly from the pattern observed in the graph

Consider an anomaly scoring function $f_a(x) \in [0,1]$, x is entity of interest
$f_a(x) \rightarrow 0$, normal
$f_a(x) \rightarrow 1$, abnormal

Challenges for Static Graph Anomalies

1. Noisy / incorrect labels
2. Lack of labelled datasets
3. Explainability / attribution

+ Inherent difficulty with working on graph data

Outline

1. introduction to anomaly detection in graphs
2. **anomaly detection in dynamic graphs**
3. laplacian change point detection for dynamic graphs
4. multi-view change point detection for dynamic graphs
5. Fast and Attributed Change Detection on Dynamic Graphs with Density of States
6.

Representing Dynamic Graphs

1. Discrete Time Dynamic Graphs: $\mathcal{G} = \{ \mathbf{G}_1, \ldots, \mathbf{G}□ \}$ (a sequence of graph snapshots)
    a. Useful for settings where there is clear boundary between timestamps: days, months, years, also the setting of this talk
2. Continuous Time Dynamic Graphs: $\mathcal{G} = \{ (s_0,d_0,t_0), (s_1,d_1,t_1), \ldots \}$, (t is ordered)
    a. Can have node / edge addition, deletion, modification events
    b. Works best for continuous time & online settings
    c. Can just be an ordered list of edges with no timestamps
    d. Could have restrictions on memory, storage, etc.

Anomaly Detection in Dynamic Graphs

- Anomalous nodes
- Anomalous edges
- Anomalous subgraphs
- Anomalous snapshots

Dynamic graphs can be directed / undirected, weighted, attributed etc. depend on the type of the graph

Challenges for Dynamic Graph Anomaly Detection

1. Temporal reasoning
2. Scalability (or streaming settings)
3. Anomaly attribution
4. Lack of labelled data / noisy labels
5. Anomalies are almost always out of distribution
6. Malicious attacks can adapt to existing methods

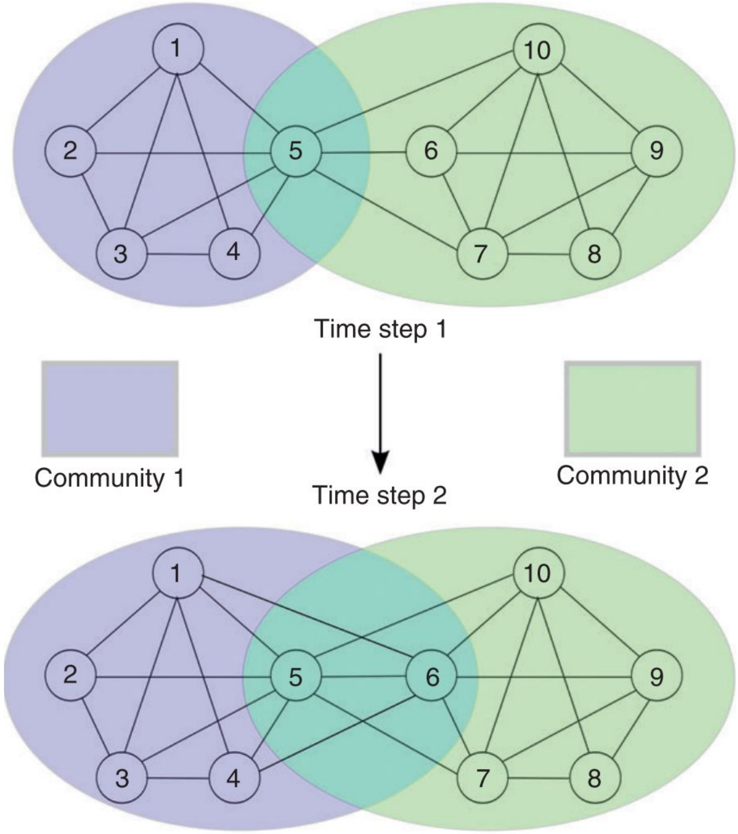
A General Approach

How to detect anomalous entities in a dynamic graph?

1. Design a scoring function or summary of the entities of interest
2. Compare such score or summary to the norm or majority in the graph
3. Output entities with abnormal scores as anomalies

Effectively designing, computing and analyzing **an anomaly score** $f_a(x)$

Anomalous nodes



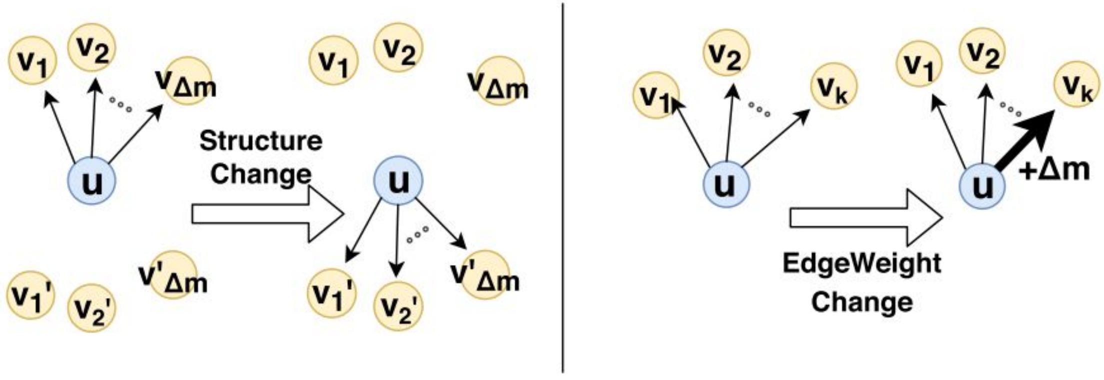
Set of nodes which have ‘irregular’ evolution when compared to other nodes

Example applications:

1. nodes that contribute most to an event in communication networks
2. nodes that switches community
3. nodes which are bots in a social network

Reference:
[Anomaly detection in dynamic networks: a survey]

Anomalous edges



(a) Structure Change (b) Edge Weight Change

Edges which have abnormal structural or weight changes
(or other types of abnormal evolution)

Example applications:

1. Email spams
2. Follower boosting
3. Denial of service attacks

Reference:
Fast and Accurate Anomaly Detection in Dynamic Graphs with a Two-Pronged Approach (KDD 2019)

Anomalous Subgraphs

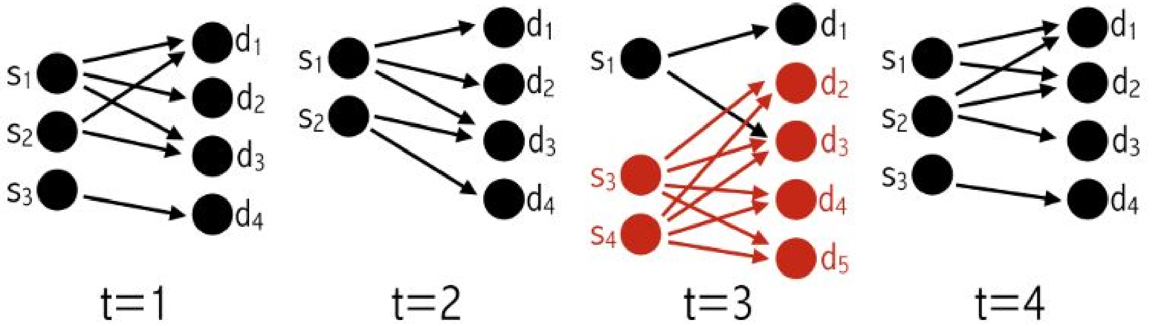


Figure 1: Sudden appearance of a dense subgraph at t=3.

Reference:
SpotLight: Detecting Anomalies in Streaming Graphs
(KDD 2018)

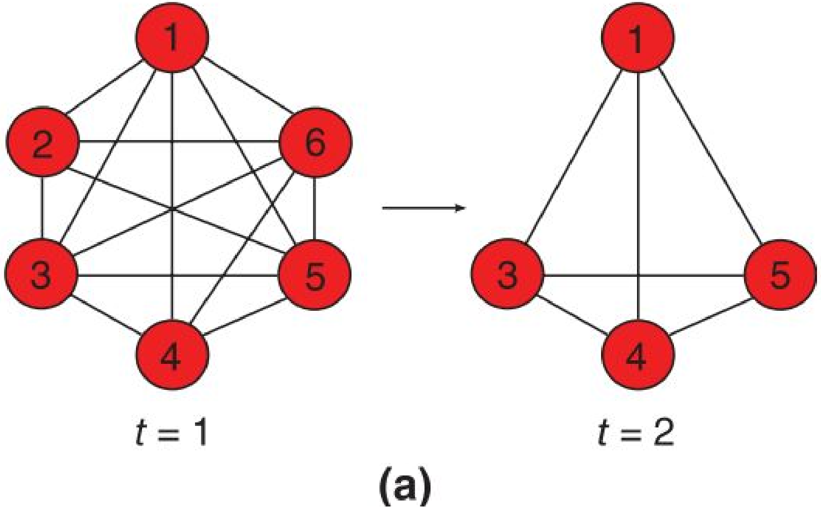
Finding anomalous evolution for **a fixed set of subgraphs**

Enumerating all possible subgraph is intractable

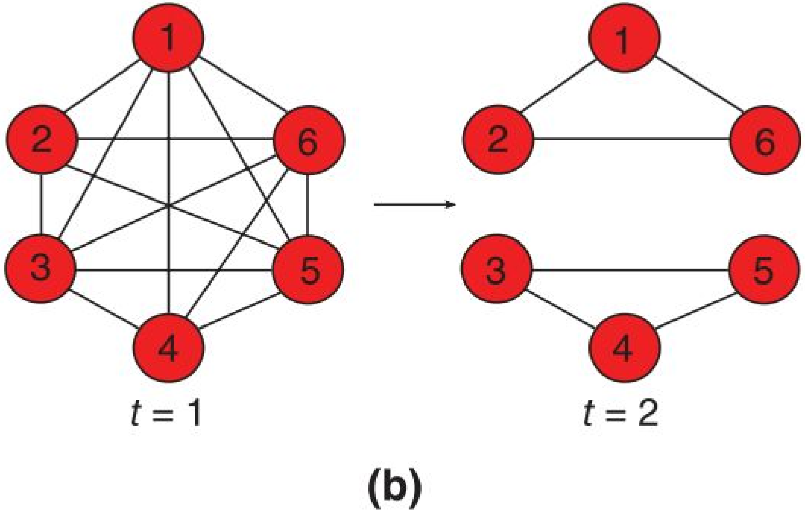
Example applications:

1. Tracking nodes of interest
2. Community splitting, merging, etc.
3. Port scans from IP-IP communication data

Anomalous Community Evolution



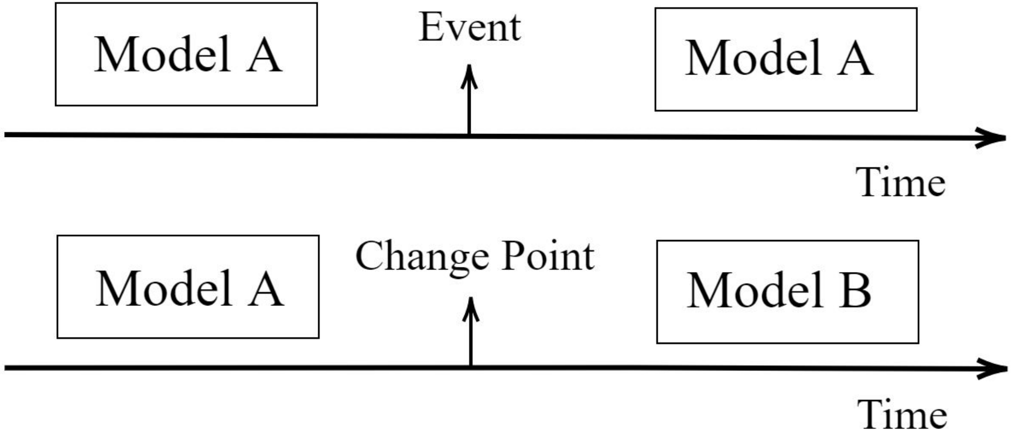
Shrinking community



Splitting community

Reference:
[Anomaly detection in dynamic networks: a survey]

Anomalous Snapshots



Identify time points where the **underlying graph generative model changes**
(change points)

or the overall **graph structure** undergoes drastic **one-time changes**
(Events)

Example Applications:

1. Traffic accidents
2. Changes in political environment
3. Events in social network

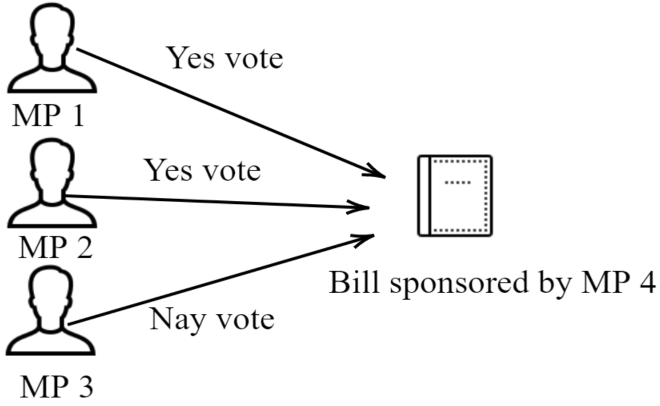
Families of Approaches

1. Community detection based
2. Minimum Description Length (MDL) and Compression based
3. Matrix / Tensor decomposition based
4. Metrics / Distance based
5. Probabilistic method / Hypothesis test based
6. Graph Neural Network (GNN) based

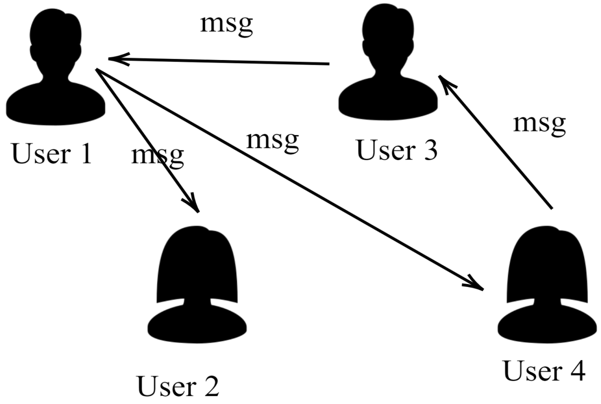
Outline

1. introduction to anomaly detection in graphs
2. anomaly detection in dynamic graphs
3. **laplacian change point detection for dynamic graphs**
4. multi-view change point detection for dynamic graphs
5. Fast and Attributed Change Detection on Dynamic Graphs with Density of States

Example Dynamic Networks



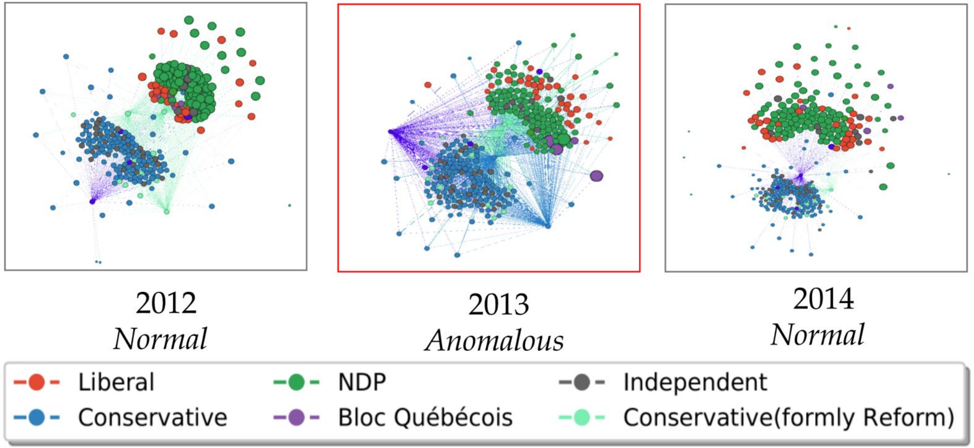
Canadian Bill Voting Network
MP - Member of Parliament



University of California, Irvine social network

Laplacian Anomaly Detection (LAD)

A spectral method for anomalous snapshot detection, presented in KDD 2020



2012
Normal

2013
Anomalous

2014
Normal

Detects the changes in Canadian Member of Parliament voting pattern. 2013 is identified as an anomalous year due to increase in cross community communication (as Justin Trudeau is elected leader of Liberal Party)

Reference; [Laplacian Change Point Detection for Dynamic Graphs]

Key Components of LAD

1. Summarize the graph snapshot at each step

   Using the **Laplacian eigenvalues σ□**

2. Compare with the norm

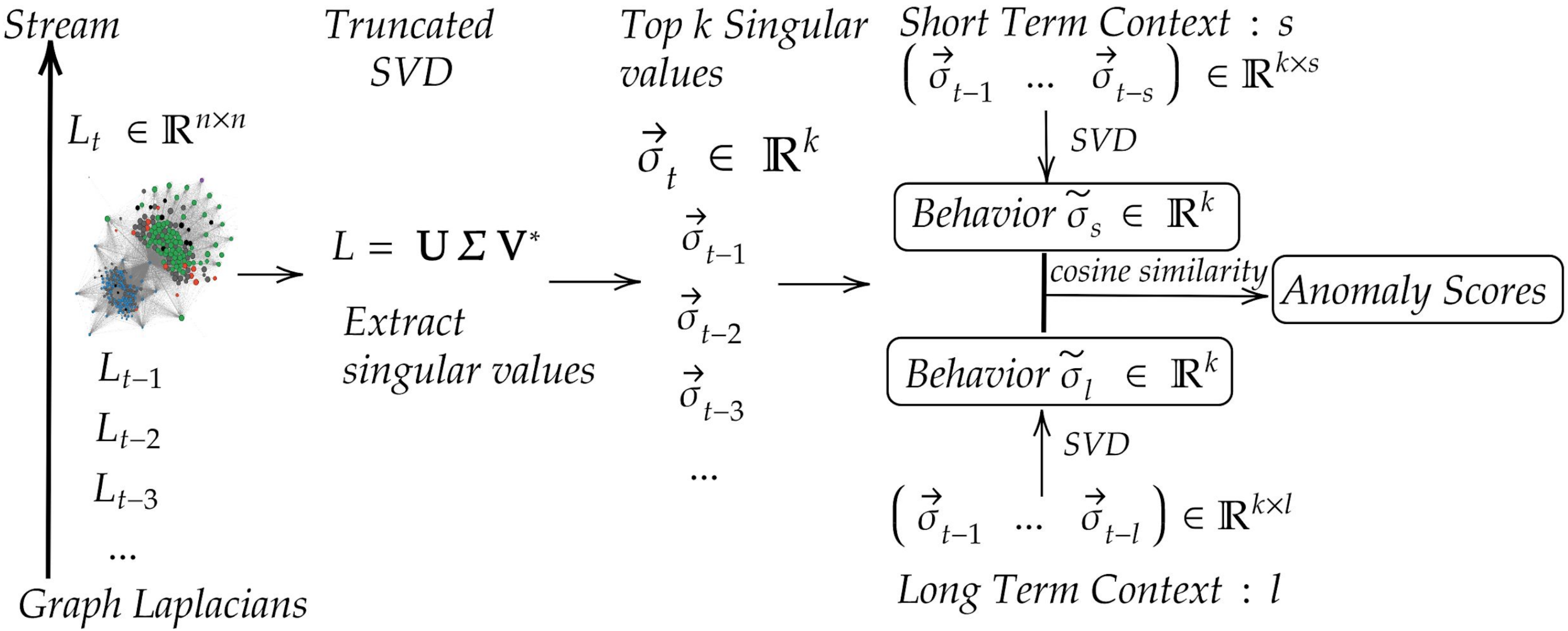
   Extract norm from a short term and a long term sliding window

3. Compute the anomaly score

   Cosine similarity between two vectors

$$cos(\theta) = \frac{\vec{u} \cdot \vec{v}}{||\vec{u}||||\vec{v}||}$$

LAD Methodology



Stream

$L_t \in \mathbb{R}^{n \times n}$

L_{t-1}

L_{t-2}

L_{t-3}

...

Graph Laplacians

Truncated SVD

$L = \mathbf{U} \boldsymbol{\Sigma} \mathbf{V}^*$

Extract singular values

Top k Singular values

$\vec{\sigma}_t \in \mathbb{R}^k$

$\vec{\sigma}_{t-1}$

$\vec{\sigma}_{t-2}$

$\vec{\sigma}_{t-3}$

...

Short Term Context : s

$\left(\vec{\sigma}_{t-1} \ \ldots \ \vec{\sigma}_{t-s} \right) \in \mathbb{R}^{k \times s}$

SVD

$\text{Behavior } \tilde{\sigma}_s \in \mathbb{R}^k$

cosine similarity

Anomaly Scores

$\text{Behavior } \tilde{\sigma}_l \in \mathbb{R}^k$

SVD

$\left(\vec{\sigma}_{t-1} \ \ldots \ \vec{\sigma}_{t-l} \right) \in \mathbb{R}^{k \times l}$

Long Term Context : l

Properties of Laplacian Eigenvalues

1. Forms a spectral signature of the graph
    a. Many connections to graph structure, connectivity and geometry
    b. Can one uniquely determine the structure of a network from the spectrum of the Laplacian?
2. node permutation invariant
3. Encodes compression loss of low rank approximations of the Laplacian
4. Corresponds to singular values in the asymmetric case

Laplacian Eigenvalues & Connectivity

$L = D - A$ for a graph G

$0 = \lambda_1 \leq \lambda_2 \leq \ldots \leq \lambda$□

$\lambda_2 \neq 0$ iff the graph is connected

# 0 eigenvalues = # of connected components

Laplacian Eigenvalues & Geometry of the Graph

some simple graph structures and their Laplacian eigenvalues:

- Fully Connected: 0, n, ..., n
- Star Graph: 0, 1, ..., n
- Cycle Graph:
    - 0, $\lambda_2 = \lambda_3 = 2 - 2\cos(2\pi / n)$, $\lambda_4 = \lambda_5 = 2 - 2\cos(4\pi / n)$, ...
- Path Graph: 0, $\lambda_{\square+1} = 2 - 2\cos(\pi k / n)$

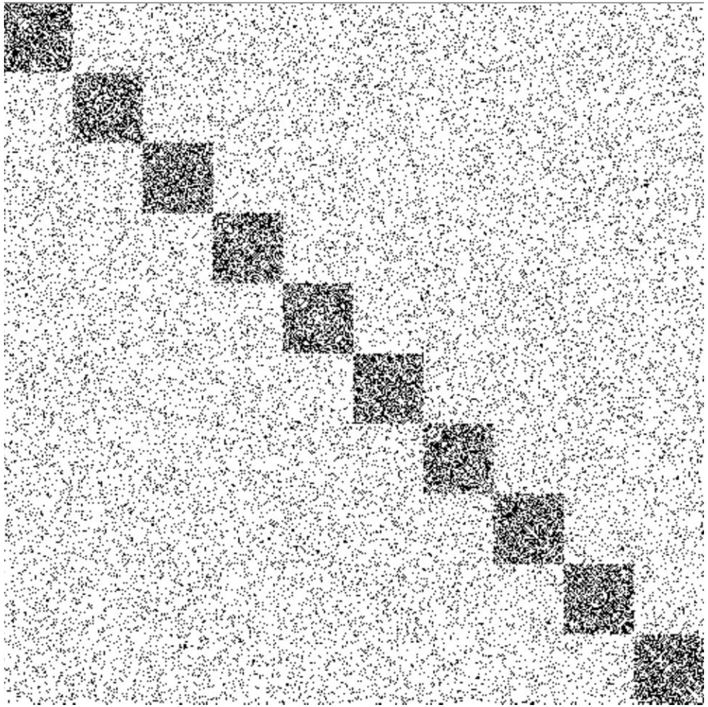
Proof and more details see Chapter 6 in
Spectral and Algebraic Graph Theory by Daniel A. Spielman

Recall: Stochastic Block Model (SBM)

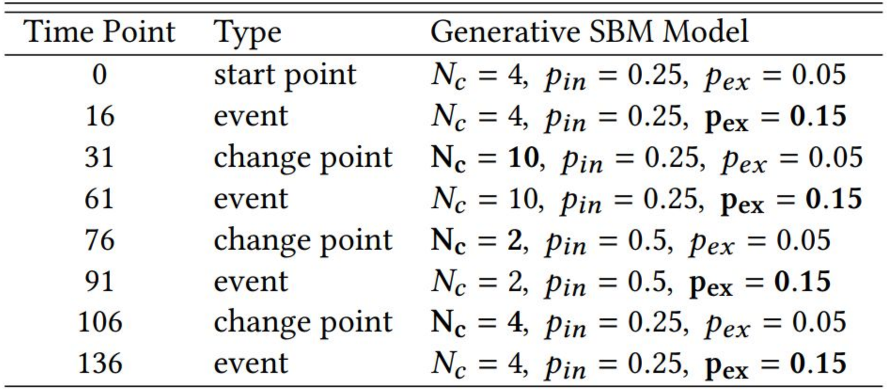
Create synthetic community structure

Parameters:

- **n:** number of nodes
- **B:** number of blocks,
  - disjoint sets that divide the n nodes
- **P:** B x B matrix
  - with probabilities per pairs of block



Synthetic Dynamic Graphs



Time Point	Type	Generative SBM Model
0	start point	$N_c = 4,\ p_{in} = 0.25,\ p_{ex} = 0.05$
16	event	$N_c = 4,\ p_{in} = 0.25,\ \mathbf{p_{ex} = 0.15}$
31	change point	$\mathbf{N_c = 10},\ p_{in} = 0.25,\ p_{ex} = 0.05$
61	event	$N_c = 10,\ p_{in} = 0.25,\ \mathbf{p_{ex} = 0.15}$
76	change point	$\mathbf{N_c = 2},\ p_{in} = 0.5,\ p_{ex} = 0.05$
91	event	$N_c = 2,\ p_{in} = 0.5,\ \mathbf{p_{ex} = 0.15}$
106	change point	$\mathbf{N_c = 4},\ p_{in} = 0.25,\ p_{ex} = 0.05$
136	event	$N_c = 4,\ p_{in} = 0.25,\ \mathbf{p_{ex} = 0.15}$

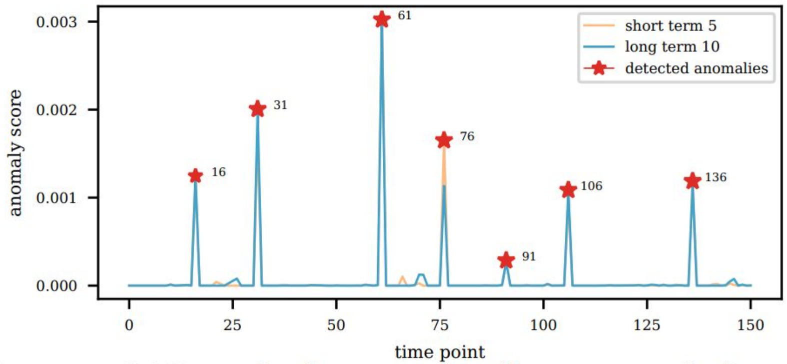
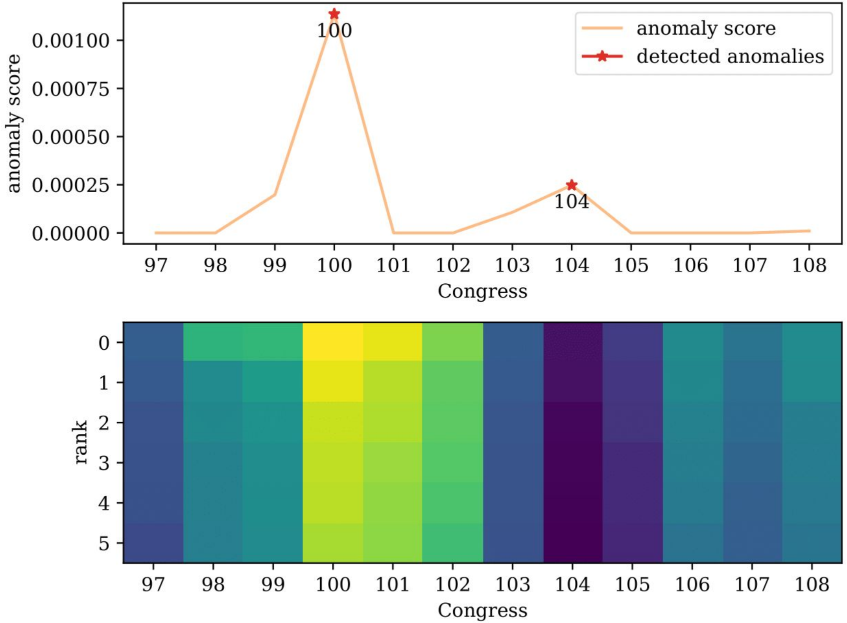


Figure 5: LAD perfectly recovers all events and change points defined in Table 4.

Change point = change in community structure in SBM

Event = sudden increase of cross community connections

Laplacian Spectrum



US Senate Co-sponsorship Network

UCI Message Network

weighted, directed social network

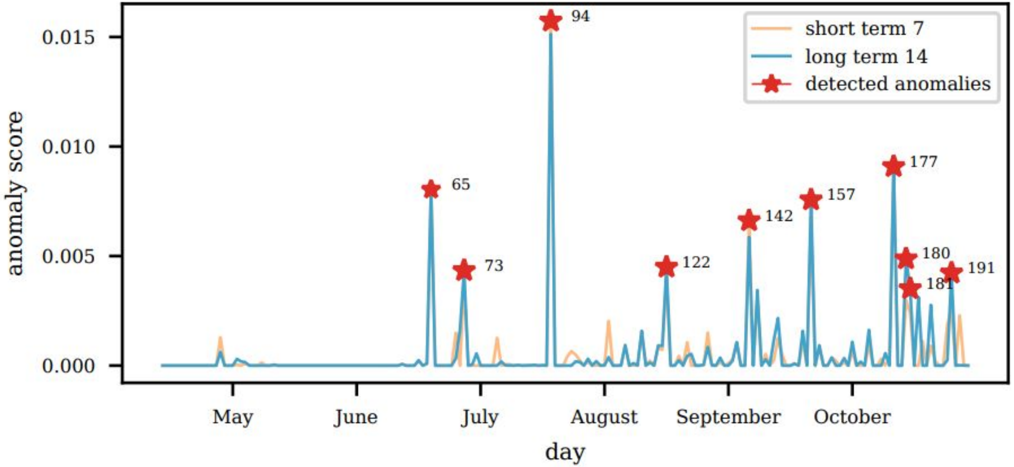


Figure 6: LAD correctly detects the end of the university spring term and one day before the start of the fall term in the UCI message dataset.

Publicly Available Data and Code

Paper link: https://dl.acm.org/doi/10.1145/3394486.3403077

Code Repository: https://github.com/shenyangHuang/LAD

All experiment is reproducible
and all dataset is in the repo if interested

Outline

1. introduction to anomaly detection in graphs
2. anomaly detection in dynamic graphs
3. laplacian change point detection for dynamic graphs
4. **multi-view change point detection for dynamic graphs**
5. Fast and Attributed Change Detection on Dynamic Graphs with Density of States

Multi-view Change Point Detection

Given a multi-view dynamic graph $\mathbb{G} = \{\mathcal{G}\square\}$ where $1 \leq t \leq T$ and
$\mathcal{G}\square = \{G_r\}$ and $1 \leq r \leq L$ where $G_r = (V, E)$
each view is a dynamic graph that describes an overall graph generative model H,
Can we detect time points in time where **H undergoes drastic changes**?

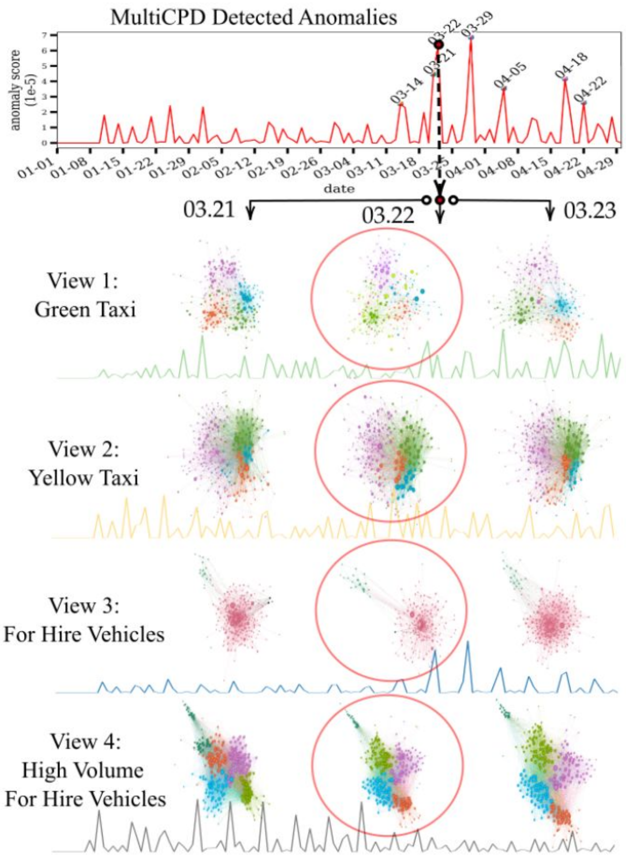
Key idea: leverage multi-view nature of the data to better **recover the underlying generative model**

Multi-view Network Examples

1. Traffic Network (same city, same external events like traffic jams)
   a. Taxis
   b. Buses
   c. Lyft / Uber
   d. Service vehicles

2. Social Network (same users, same relationships)
   a. Facebook social network
   b. Twitter follower network
   c. Instagram follower network
   d. Text chatting network

MultiCPD: a multi-view extension of LAD



New York City Taxi dataset

From Jan 2020 to Apr 2020

Each view is a different type of taxi or for hire vehicle service

03.22 is the start of the New York on Pause program

How to merge information from multiple views?

1. Aggregate the anomaly scores
   a. Still carry over noise from individual views
   b. One view could dominate the others
   c. Implemented as naive baseline: maxLAD and meanLAD
2. Aggregate the signature vectors (our approach)
   a. Merge the Laplacian eigenvalues from each view
   b. Compute an aggregated overall view
   c. Can reduce noise from individual views

Key Component of MultiCPD

1. Merging signature vectors from different views via **scalar power mean**

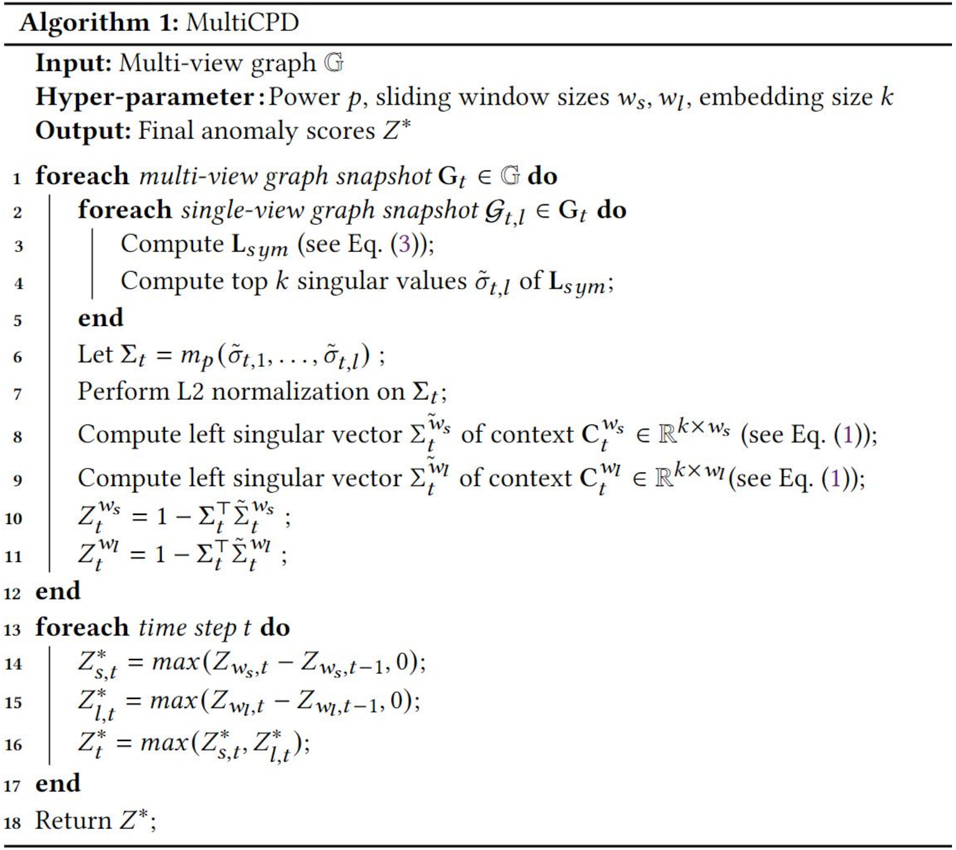
$$m_p(x_1, \ldots, x_m) = \left(\frac{1}{m} \sum_{i=1}^{m} x_i^p \right)^{\frac{1}{p}} \qquad (2)$$

$$\Sigma_s = (m_p(\lambda_{11}, \ldots, \lambda_{1m}), \ldots, m_p(\lambda_{n1}, \ldots, \lambda_{nm})) \qquad (3)$$

2. Using the normalized Laplacian matrix

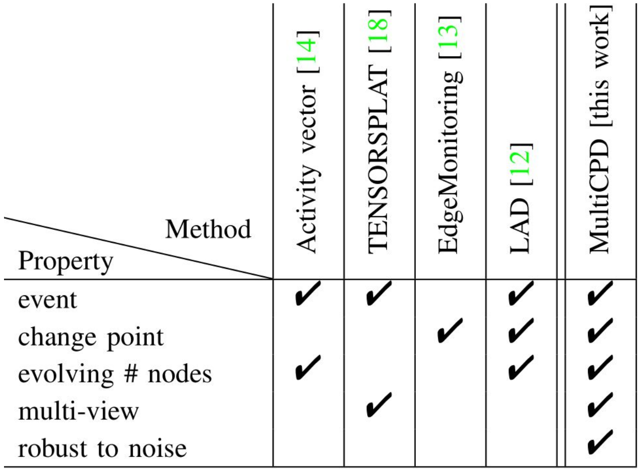
$$\mathbf{L}_{sym} = \mathbf{D}^{-\frac{1}{2}} \mathbf{L} \mathbf{D}^{-\frac{1}{2}} = \mathbf{I} - \mathbf{D}^{-\frac{1}{2}} \mathbf{A} \mathbf{D}^{-\frac{1}{2}} \qquad (1)$$

Algorithm of MultiCPD



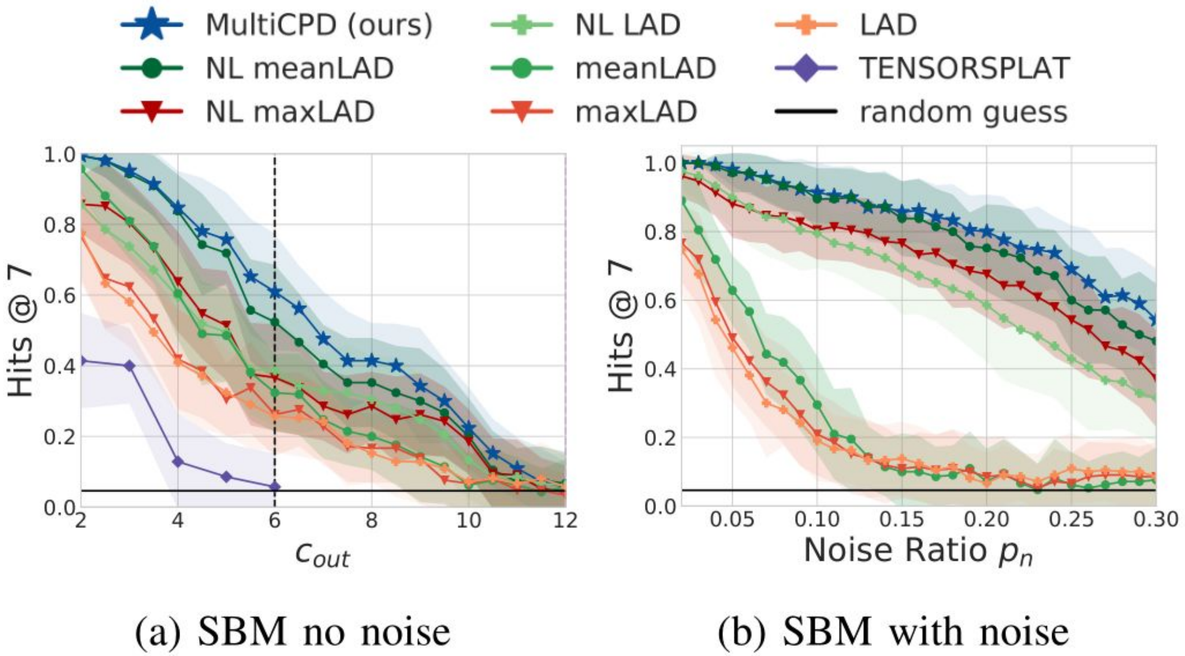
Algorithm 1: MultiCPD

```
Input: Multi-view graph $\mathbb{G}$
Hyper-parameter: Power $p$, sliding window sizes $w_s$, $w_l$, embedding size $k$
Output: Final anomaly scores $Z^*$
foreach multi-view graph snapshot $\mathbf{G}_t \in \mathbb{G}$ do
    foreach single-view graph snapshot $\mathcal{G}_{t,l} \in \mathbf{G}_t$ do
        Compute $\mathbf{L}_{sym}$ (see Eq. (3));
        Compute top $k$ singular values $\tilde{\sigma}_{t,l}$ of $\mathbf{L}_{sym}$;
    end
    Let $\Sigma_t = m_p(\tilde{\sigma}_{t,1}, \ldots, \tilde{\sigma}_{t,l})$ ;
    Perform L2 normalization on $\Sigma_t$;
    Compute left singular vector $\tilde{\Sigma}_t^{w_s}$ of context $\mathbf{C}_t^{w_s} \in \mathbb{R}^{k \times w_s}$ (see Eq. (1));
    Compute left singular vector $\tilde{\Sigma}_t^{w_l}$ of context $\mathbf{C}_t^{w_l} \in \mathbb{R}^{k \times w_l}$(see Eq. (1));
    $Z_t^{w_s} = 1 - \Sigma_t^\top \tilde{\Sigma}_t^{w_s}$ ;
    $Z_t^{w_l} = 1 - \Sigma_t^\top \tilde{\Sigma}_t^{w_l}$ ;
end
foreach time step $t$ do
    $Z_{s,t}^* = max(Z_{w_s,t} - Z_{w_s,t-1}, 0)$;
    $Z_{l,t}^* = max(Z_{w_l,t} - Z_{w_l,t-1}, 0)$;
    $Z_t^* = max(Z_{s,t}^*, Z_{l,t}^*)$;
end
Return $Z^*$;
```



Property \ Method	Activity vector [14]	TENSORSPLAT [18]	EdgeMonitoring [13]	LAD [12]	MultiCPD [this work]
event	✓	✓		✓	✓
change point			✓	✓	✓
evolving # nodes	✓			✓	✓
multi-view		✓			✓
robust to noise					✓

Multi-view Data Improves Performance



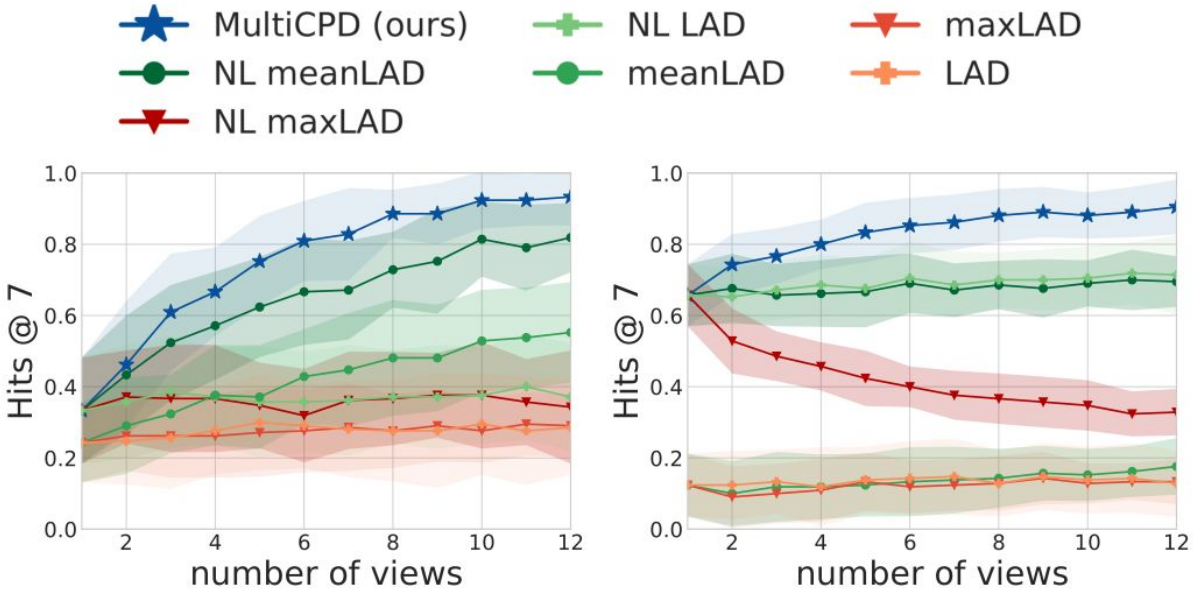
(a) SBM no noise

(b) SBM with noise

Increasing difficulty, only change points

Increasing noise, event and change point

Increasing Number of Views



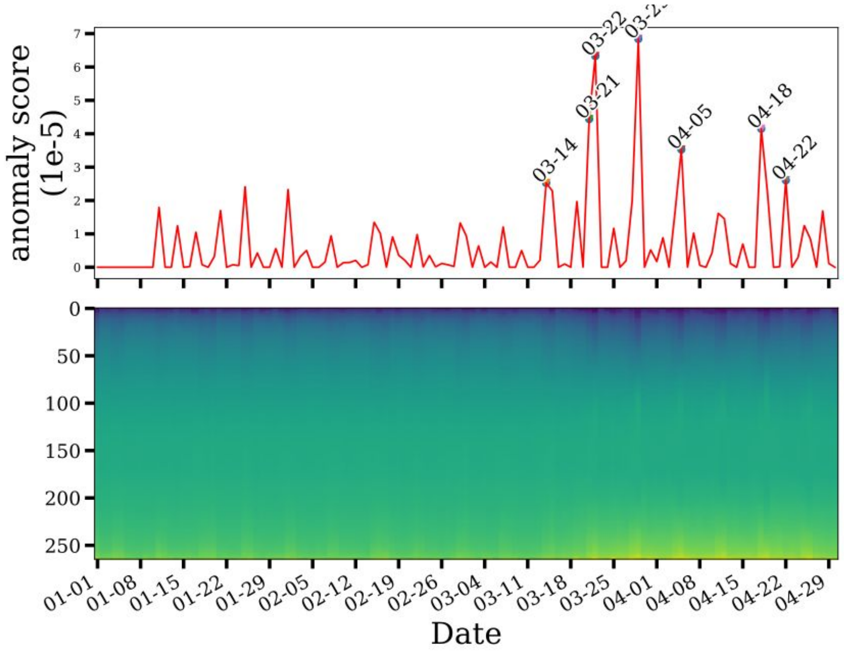
(a) SBM

(b) BA

Only change points

increasing m

NYC Taxi Dataset 2020



7-Mar-20	State of Emergency declared in New York State
8-Mar-20	Issued guidelines relating to public transit
12-Mar-20	Events with more than 500 attendees are cancelled/postponed
13-Mar-20[o]	National State of Emergency declared
16-Mar-20	NYC public schools close
17-Mar-20	NYC bars and restaurants can only operate by delivery
22-Mar-20[†]	"NYS on Pause Program" begins, all non-essential workers must stay home
28-Mar-20[o]	All non-essential construction halted
6-Apr-20[o]	Extension of of stay-at-home order and school closures
16-Apr-20	Extension of of stay-at-home order and school closures
30-Apr-20	Subway ceases to operate during early hours

Outline

1. introduction to anomaly detection in graphs
2. anomaly detection in dynamic graphs
3. laplacian change point detection for dynamic graphs
4. multi-view change point detection for dynamic graphs
5. **Fast and Attributed Change Detection on Dynamic Graphs with Density of States**

How to scale to large dynamic networks?

- Computing Laplacian eigenvalues O(N^3)

Difficult to scale to large graphs with millions of nodes

- Approximating spectral density O (|V| + |E|)

Much more scalable

Fast and efficient approximation

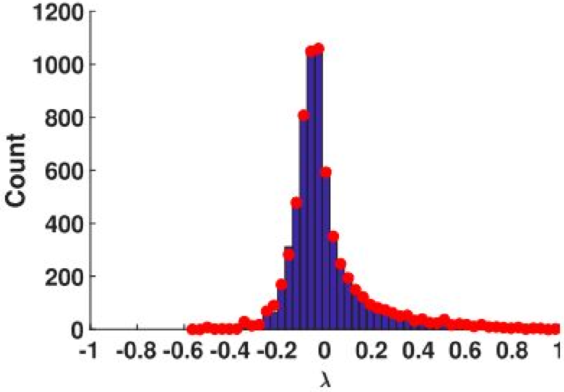
Losses information about exact values of eigenvalues in most cases

What is Density of States (Spectral Density)

$$\mu(\lambda) = \frac{1}{N}\sum_{i=1}^{N}\delta(\lambda - \lambda_i), \quad \int f(\lambda)\mu(\lambda) = \text{trace}(f(H)) \qquad (1)$$

1. Normalize the range of Laplacian eigenvalues
2. Find how many eigenvalues fall into each bin / interval

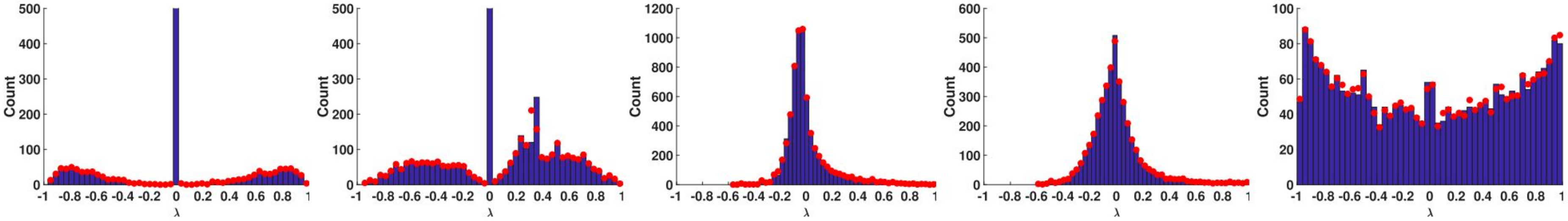
Computed by an efficient approximation method named Network Density of States or just DOS



(c) Marvel Characters Network

Reference: [Network Density of States]

Spectral Density of different networks



(a) Erdős Collaboration Network

(b) Autonomous System Network (1999)

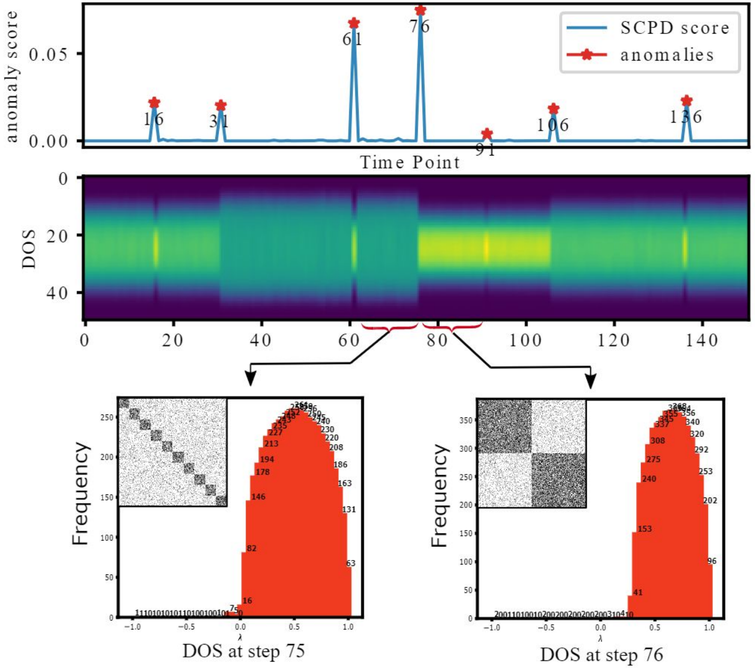
(c) Marvel Characters Network

(d) Facebook Ego Networks

(e) Minnesota Road Network

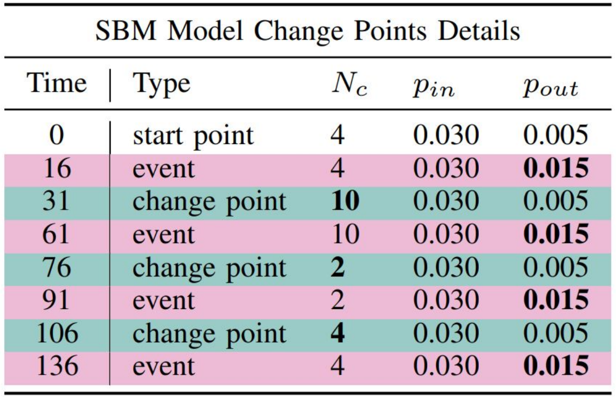
Reference: Network Density of States

Fast and Attributed Change Detection on Dynamic Graphs with Density of States



SBM model with 10 communities

SBM model with 2 communities



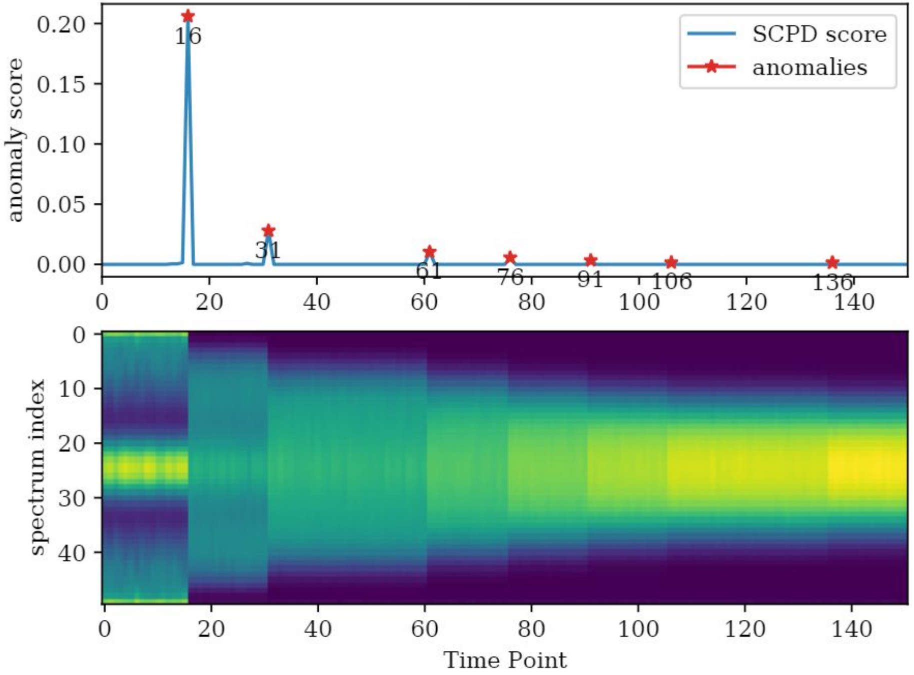
Time	Type	N_c	p_{in}	p_{out}
0	start point	4	0.030	0.005
16	event	4	0.030	**0.015**
31	change point	**10**	0.030	0.005
61	event	10	0.030	**0.015**
76	change point	**2**	0.030	0.005
91	event	2	0.030	**0.015**
106	change point	**4**	0.030	0.005
136	event	4	0.030	**0.015**

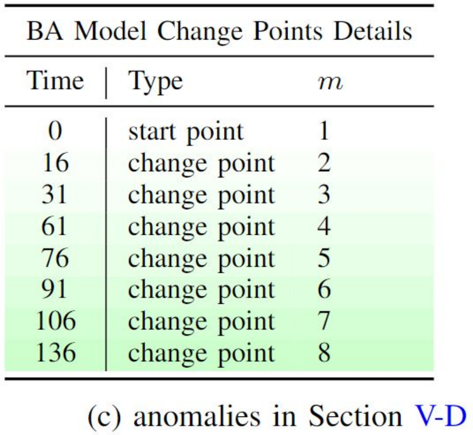
SBM Model Change Points Details

(a) anomalies in Section V-B

Prior version Reference: Scalable Change Point Detection for Dynamic Graphs

Change in DOS for BA model





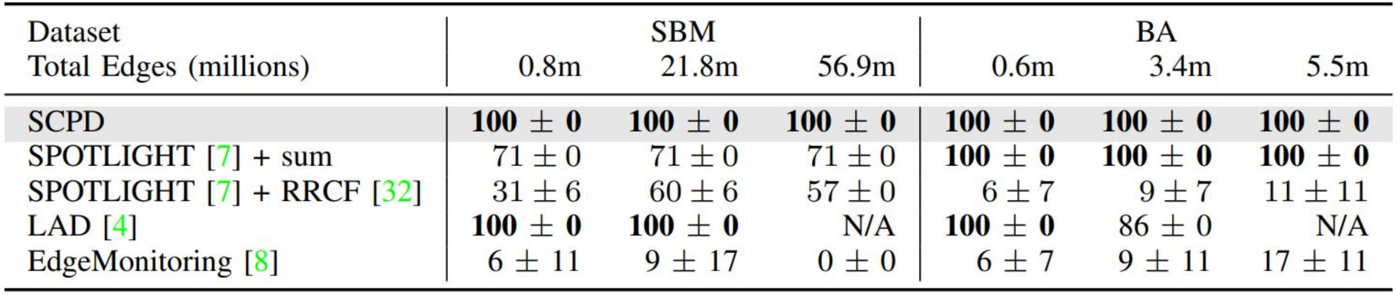
BA Model Change Points Details		
Time	Type	m
0	start point	1
16	change point	2
31	change point	3
61	change point	4
76	change point	5
91	change point	6
106	change point	7
136	change point	8

(c) anomalies in Section V-D

Figure 3: SCPD perfectly recovers all events and change points for the BA model, experiment setup explained in Table II.

Increase in m, number of edges attached from a new node to an existing node

Performance Comparison



Dataset	SBM			BA		
Total Edges (millions)	0.8m	21.8m	56.9m	0.6m	3.4m	5.5m
SCPD	**100 ± 0**	**100 ± 0**	**100 ± 0**	**100 ± 0**	**100 ± 0**	**100 ± 0**
SPOTLIGHT [7] + sum	71 ± 0	71 ± 0	71 ± 0	**100 ± 0**	**100 ± 0**	**100 ± 0**
SPOTLIGHT [7] + RRCF [32]	31 ± 6	60 ± 6	57 ± 0	6 ± 7	9 ± 7	11 ± 11
LAD [4]	**100 ± 0**	**100 ± 0**	N/A	**100 ± 0**	86 ± 0	N/A
EdgeMonitoring [8]	6 ± 11	9 ± 17	0 ± 0	6 ± 7	9 ± 11	17 ± 11

SPCD Outperforming other methods in synthetic experiments

Computational Complexity: O(E)

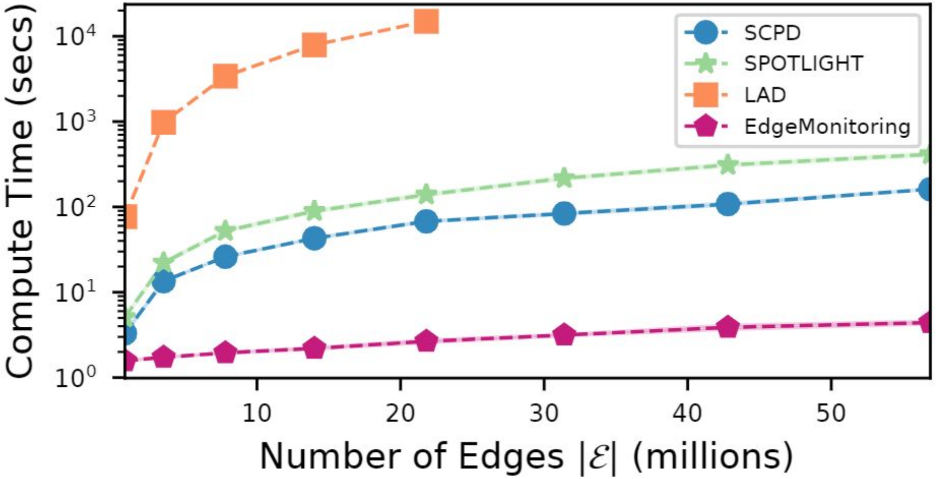


Figure 4: Compute time comparison between different methods on the SBM experiments with varying number of edges.

MAG-History Co-authorship Network

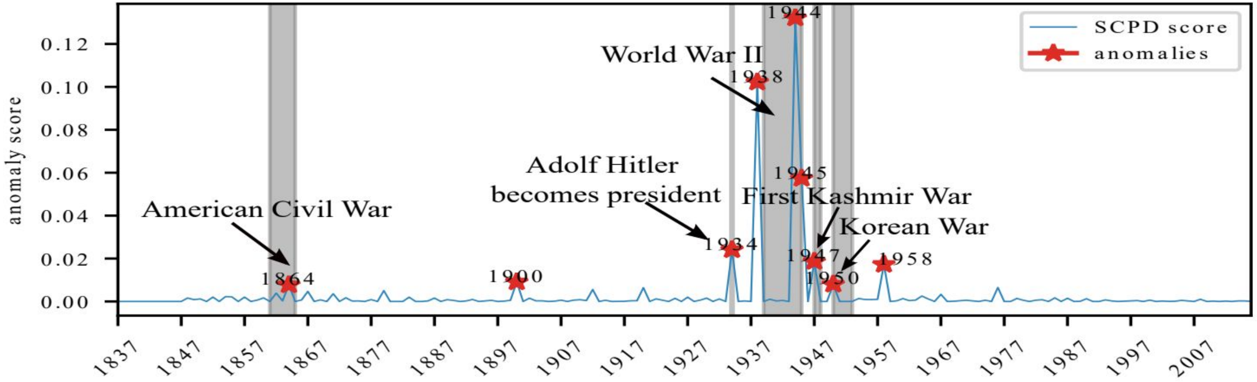


Figure 5: SCPD detects various years corresponding to historical events from the MAG-History dataset.

Co-authorship network in the history community

Attribute Changes in Flight Network

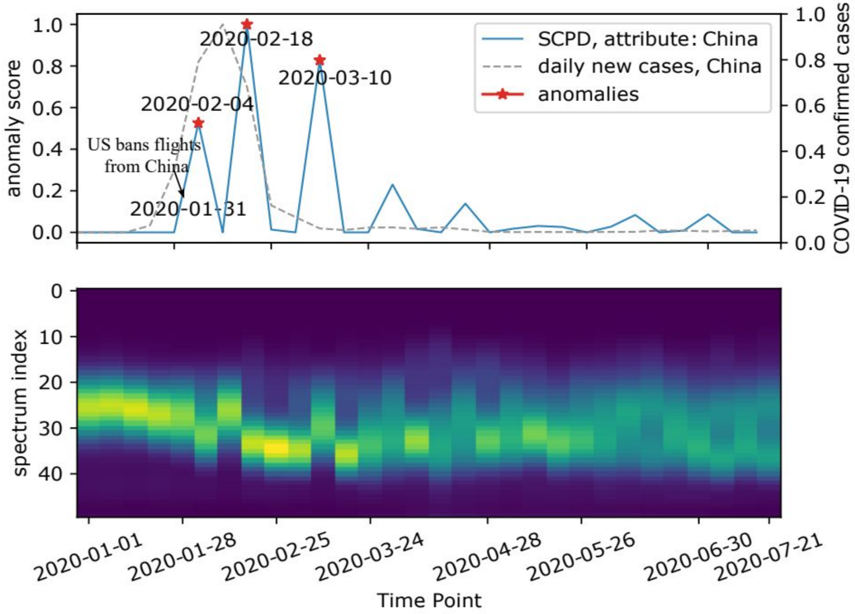


Figure 7: SCPD detects closure of flights routes to China due to covid interventions at beginning of Feb 2020. The anomaly score and case numbers are normalized to $[0, 1]$.

Attributes are the country of the airport (node)

Detects flight route closure for chinese airports at beginning of COVID-19 pandemic, Feb 2020

Thanks for Listening!

If you have any questions, feel free to reach out!

shenyang.huang@mail.mcgill.ca or on slack

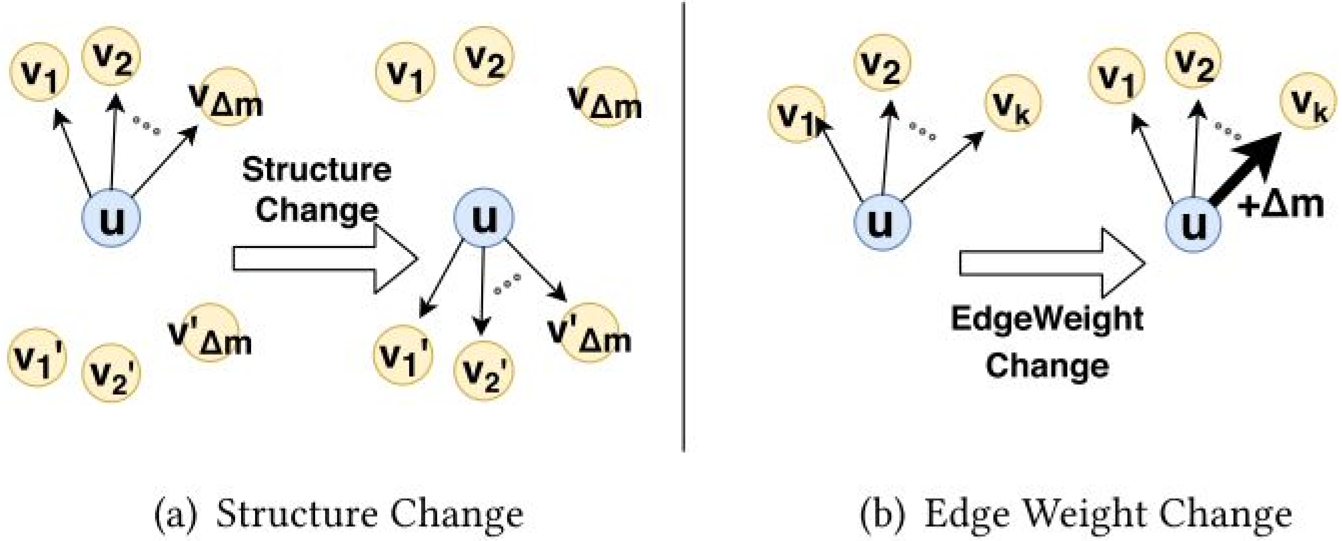
Tips for Research

1. Start with a task of interest
   a. Anomalous nodes / edges / subgraphs or snapshots
2. Find some recent relevant papers
3. Examine how the paper fits the general approach
   a. What is the summary used?
   b. How to compare with normal / expected behavior?
   c. What is the anomaly score?
4. Identify some insights & intuition
5. Find some procedures which you can improve
   a. Better scalability? Better explainability? Better performance?

Bonus Topics

1. AnomRank: an edge anomaly detection method
2. SpotLight: a subgraph anomaly detection method

AnomRank



(a) Structure Change

(b) Edge Weight Change

Structural anomaly
ANOMALYS

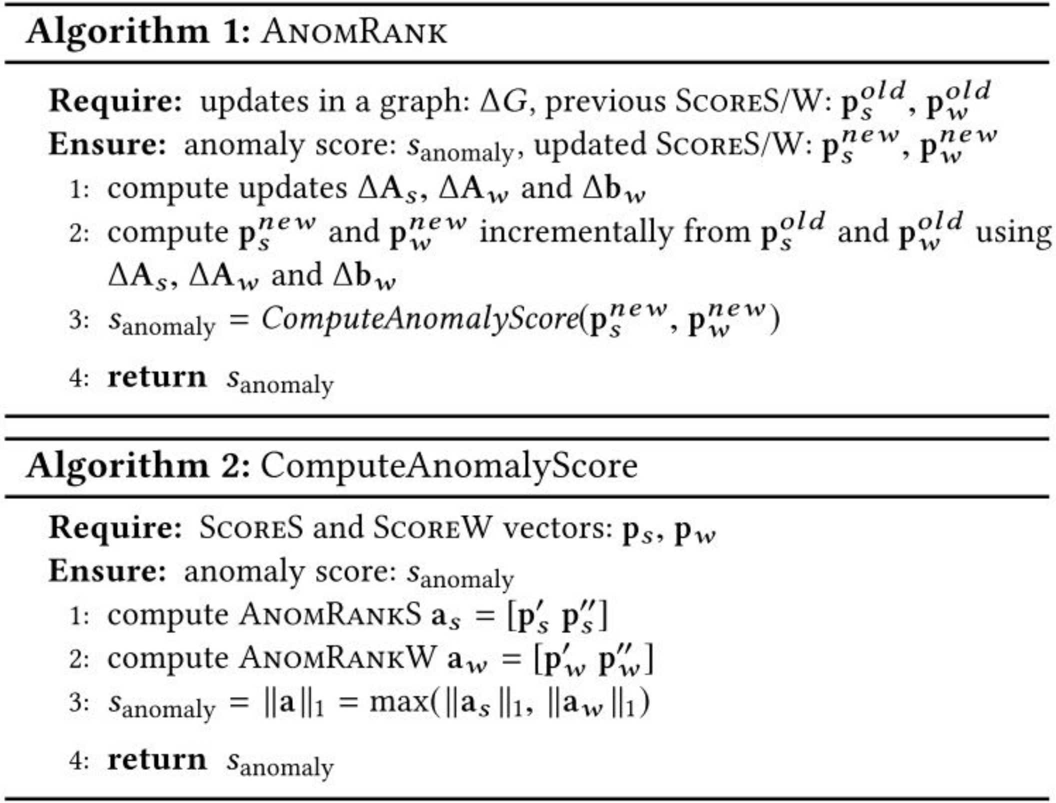
Weight anomaly
ANOMALYW

Reference:
[Fast and Accurate Anomaly Detection in Dynamic Graphs with a Two-Pronged Approach]

ANOMRANK Overview

1. Compute a node score for ANOMALYS and ANOMALYW
    a. General approach step 1: scoring function or summary
    b. Here the node score is chosen to be PageRank and weighted extension of PageRank
2. Look at 1st and 2nd order derivatives of node scores
    a. General approach step 2: how it is different from the norm
    b. Abrupt gains or losses are reflected in the derivatives
3. Compute an anomaly score for each node
    a. General approach step 3: compute the anomaly score
    b. Rank the anomalous node and edges based on the anomaly score

ANOMRANK Algorithm



Algorithm 1: AnomRank

```
Require: updates in a graph: $\Delta G$, previous ScoreS/W: $\mathbf{p}_s^{old}$, $\mathbf{p}_w^{old}$
Ensure: anomaly score: $s_{\text{anomaly}}$, updated ScoreS/W: $\mathbf{p}_s^{new}$, $\mathbf{p}_w^{new}$
  compute updates $\Delta\mathbf{A}_s$, $\Delta\mathbf{A}_w$ and $\Delta\mathbf{b}_w$
  compute $\mathbf{p}_s^{new}$ and $\mathbf{p}_w^{new}$ incrementally from $\mathbf{p}_s^{old}$ and $\mathbf{p}_w^{old}$ using
      $\Delta\mathbf{A}_s$, $\Delta\mathbf{A}_w$ and $\Delta\mathbf{b}_w$
  $s_{\text{anomaly}} = \mathit{ComputeAnomalyScore}(\mathbf{p}_s^{new}, \mathbf{p}_w^{new})$
  return $s_{\text{anomaly}}$
```

Algorithm 2: ComputeAnomalyScore

```
Require: ScoreS and ScoreW vectors: $\mathbf{p}_s$, $\mathbf{p}_w$
Ensure: anomaly score: $s_{\text{anomaly}}$
  compute AnomRankS $\mathbf{a}_s = [\mathbf{p}'_s \; \mathbf{p}''_s]$
  compute AnomRankW $\mathbf{a}_w = [\mathbf{p}'_w \; \mathbf{p}''_w]$
  $s_{\text{anomaly}} = \|\mathbf{a}\|_1 = \max(\|\mathbf{a}_s\|_1, \|\mathbf{a}_w\|_1)$
  return $s_{\text{anomaly}}$
```

Pros:

1. Fast, linear to # edges
2. Anomaly attribution
3. Works well on ENRON email and DARPA (network intrusion detection)

Cons:

1. Look at sudden changes (compare with immediate past graph)
2. Can miss global changes

SPOTLIGHT

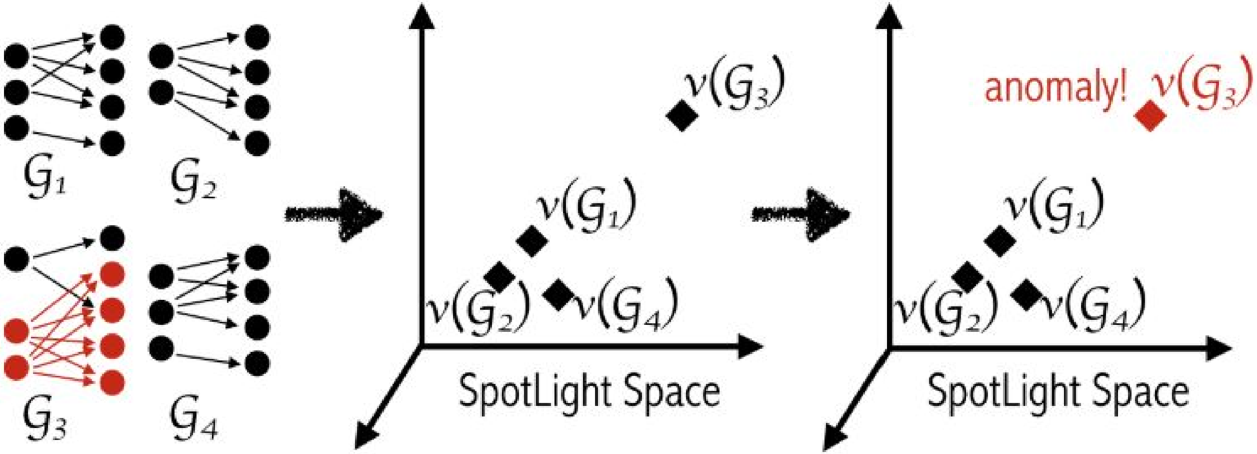


Figure 2: Overview of SpotLight

Reference:
[SpotLight: Detecting Anomalies in Streaming Graphs]

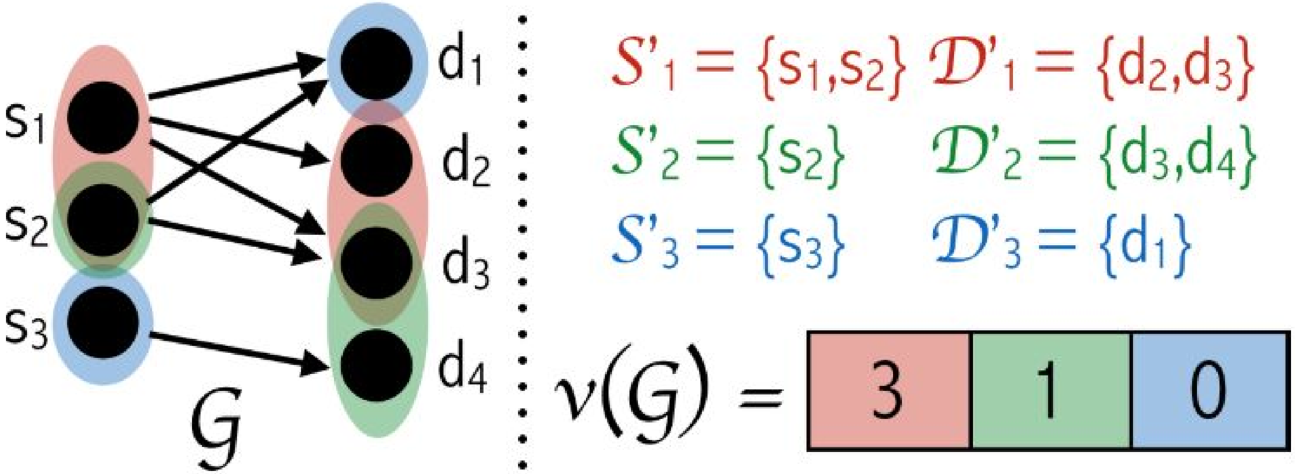
How SPOTLIGHT Works

1. Randomly select k subgraphs (fixed over time)
2. Report the total weight of edges in each subgraph as individual dimensions in the SPOTLIGHT Sketch
    a. General Approach step 1: summary of each time step
3. Report anomaly in the SPOTLIGHT Sketch space
    a. General Approach step 2 and 3

Key: use dictionaries for sublinear memory and fast run time

Assumes nodes don’t disappear

SPOTLIGHT Sketch Space



Reference:
[SpotLight: Detecting Anomalies in Streaming Graphs]

Laplacian Eigenvalue Slides

Star Graph

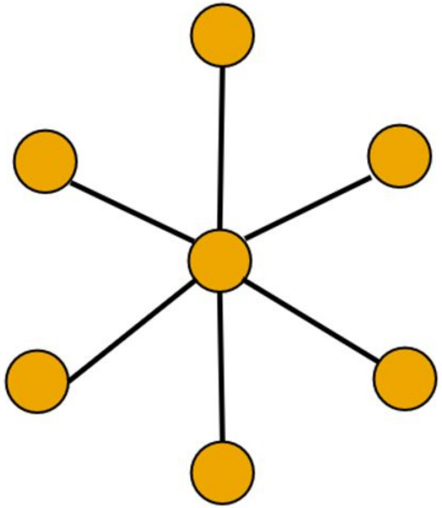
What would be the Laplacian eigenvalues ?

Hint: let there be **n nodes**,

1 hub with degree n,

Other nodes have degree 1

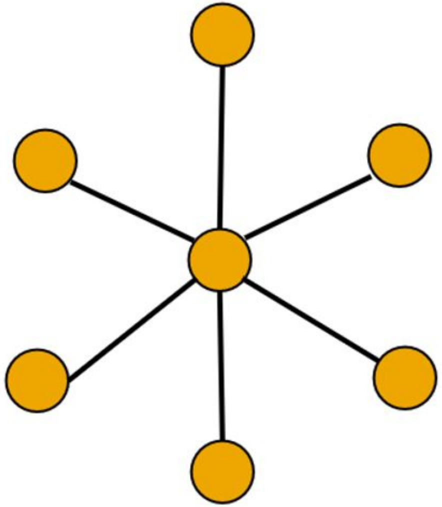
$$0 = \lambda_1 \leq \lambda_2 \leq \ldots \leq \lambda_n$$



Spectral Graph Theory: Star Graph

$$\lambda_1 = 0$$

$$\lambda_2 = ?$$

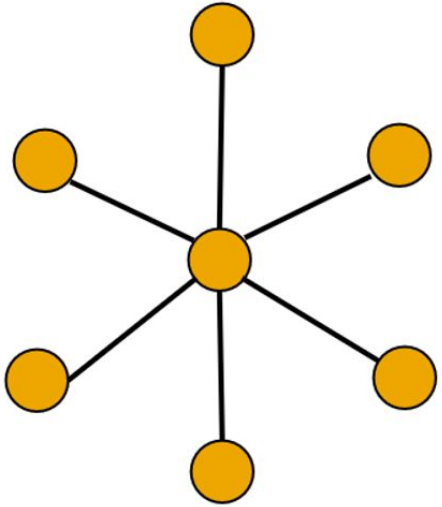


Spectral Graph Theory: Star Graph

$$\lambda_1 = 0$$

$$\lambda_2 = \lambda_3 = ... = \lambda_{n-1} = 1$$

$$\lambda_n = ?$$

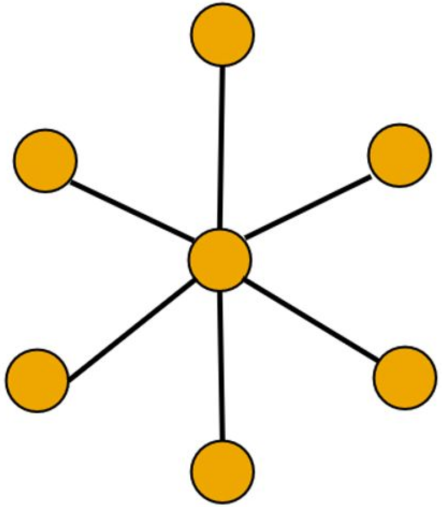


Spectral Graph Theory: Star Graph

$$\lambda_1 = 0$$

$$\lambda_2 = \lambda_3 = \ldots = \lambda_{n-1} = 1$$

$$\lambda_n = n$$



Proof and more details see Chapter 6 in

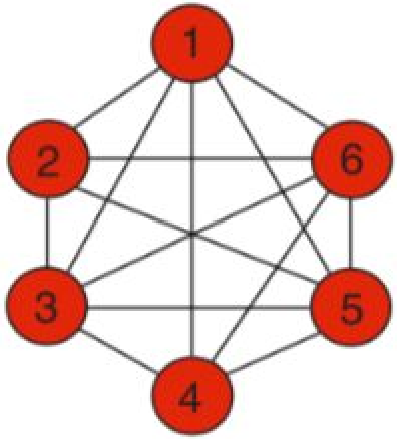
Spectral and Algebraic Graph Theory by Daniel A. Spielman

Fully Connected Graph

What would be the Laplacian eigenvalues of a fully connected graph?

Hint: let there be **n nodes**, we know that this is the highest possible connectivity

$$0 = \lambda_1 \leq \lambda_2 \leq ... \leq \lambda_n$$

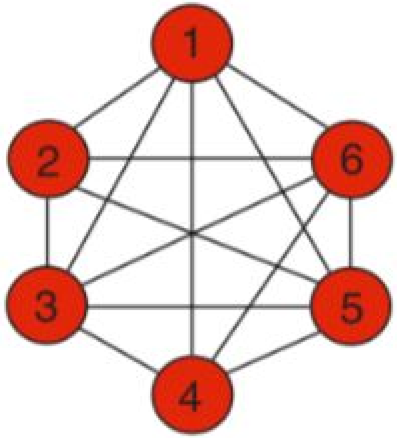


Spectral Graph Theory: Fully Connected Graph

The eigenvalues would be

$$\lambda_1 = 0$$

$$n = \lambda_2 = \lambda_3 = \ldots = \lambda_n$$



Proof and more details see Chapter 6 in

Spectral and Algebraic Graph Theory by Daniel A. Spielman