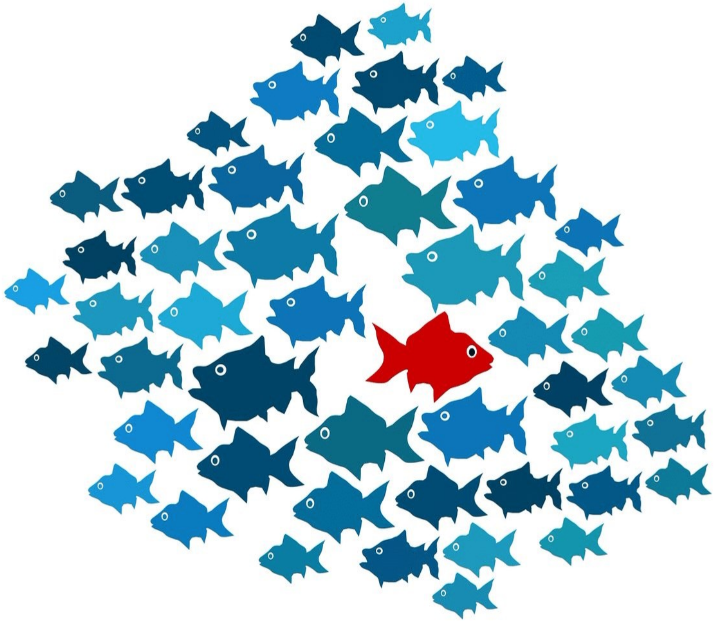
Social Network Analysis

ANOMALY DETECTION IN NETWORKS

Anomaly Detection



https://thedatascientist.com/anomaly-detection-why-you-need-it/

What is anamoly detection?

- ❑Identification of rare items, events or observations which raise suspicions by differing significantly from the majority of the data

- ❑An anomaly can be defined as a pattern or behavior that deviates from the expected trend

What are the other words used for anamoly?

- ❑Referred to as outliers, novelties, noise, deviations, exceptions, etc. depending on the nature of application

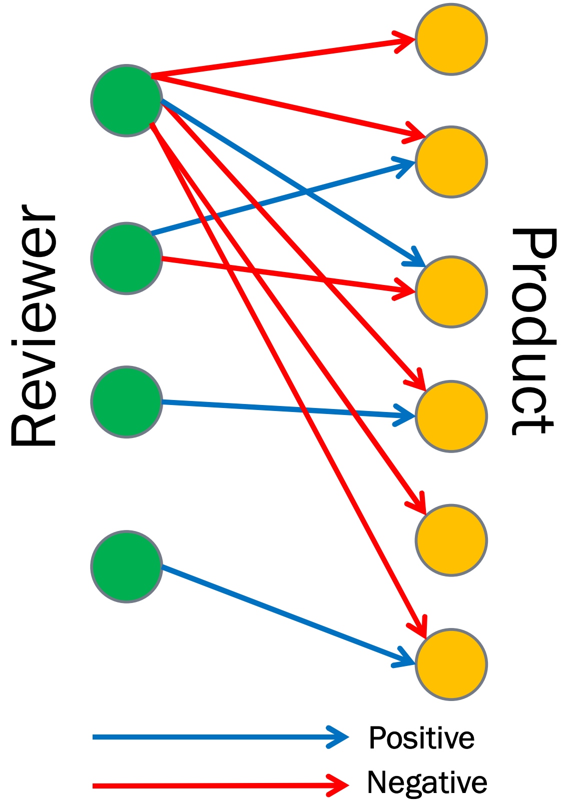
Why do we need to detect anamoly?

- ❑Anomalous data can indicate critical incidents
  - ❑A technical glitch, or
  - ❑A potential opportunities!!

Are all anamolies bad. => false Why?

- ❑Anomalies aren't categorically good or bad; they're just deviations from the expected value for a metric at a given point in time

Anomaly Detection:
Outlier-based vs. Network-based



- In outlier-based anomaly detection;
  - dataset is generally mapped into a feature space and processed further for anomalies

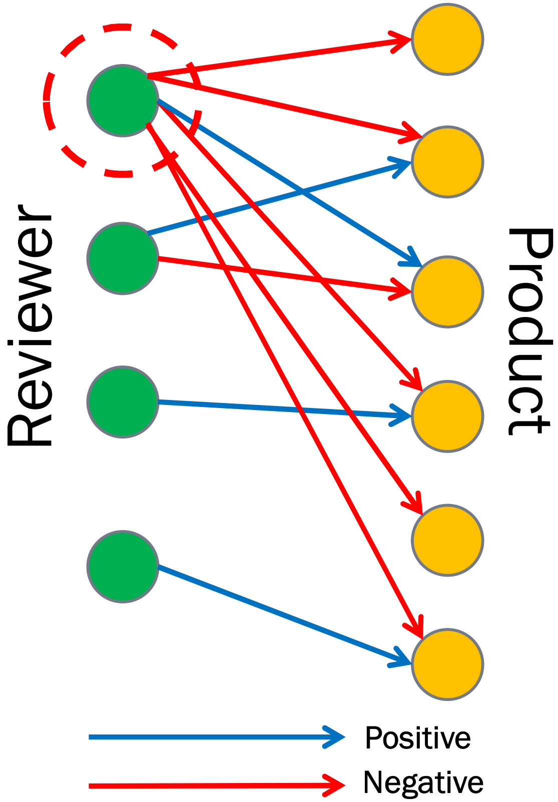
    what is the fundamental difference of detecting an anamoly for both of these kind of approaches?

- In network-based anomaly detection,
  - the dataset is mapped into a network; the anomalies are detected based on this network dataset
  - the network captures inter-dependencies among nodes

    how does the network help in the detecting the anamoly.

- Difficult to generate a concrete definition of what constitutes an anomaly

Network-based Anomaly Detection



- Some scenarios involving anomalous behavior
  - a node having a rare combination of features in an attributed network kaik alag j vastu thi banelu hoi.
  - isolated nodes located far away from a majority of nodes in the network neha jeva nodes ke je aekla j chale ....
  - nodes that are surprising and are not adhere to the underlying distribution of the network model, etc.

amuk to nodes j alag hoi...gungun jeva => je loko ni varta pan alg hoi. = they are not aena aju baju na loko(not adhere to the underlying distribution of the network model).

- The encircled reviewer's behavior seems anomalous Kem encricled node no behavior is anomalous?
  - She has given all but one review negative!!!
  - Clear deviation from behavior of other nodes in the network
  - Might be a fake/paid reviewer

Lack of Labelled Datasets: There's a scarcity of datasets with clear labels distinguishing between normal and anomalous data points.

Infeasibility of Manual Labelling: It's impractical to manually label data points due to the vast size of real-world networks and their dynamic growth.

Enormous and Dynamic Networks: Real-world networks are huge and constantly changing, making it difficult to keep track of anomalies.

Complex Metadata: Network entities come with a wealth of metadata, represented as intricate feature vectors, complicating anomaly detection.

Subjectivity in Interpretation: Human interpretation of anomalies varies, introducing subjectivity into the detection process.

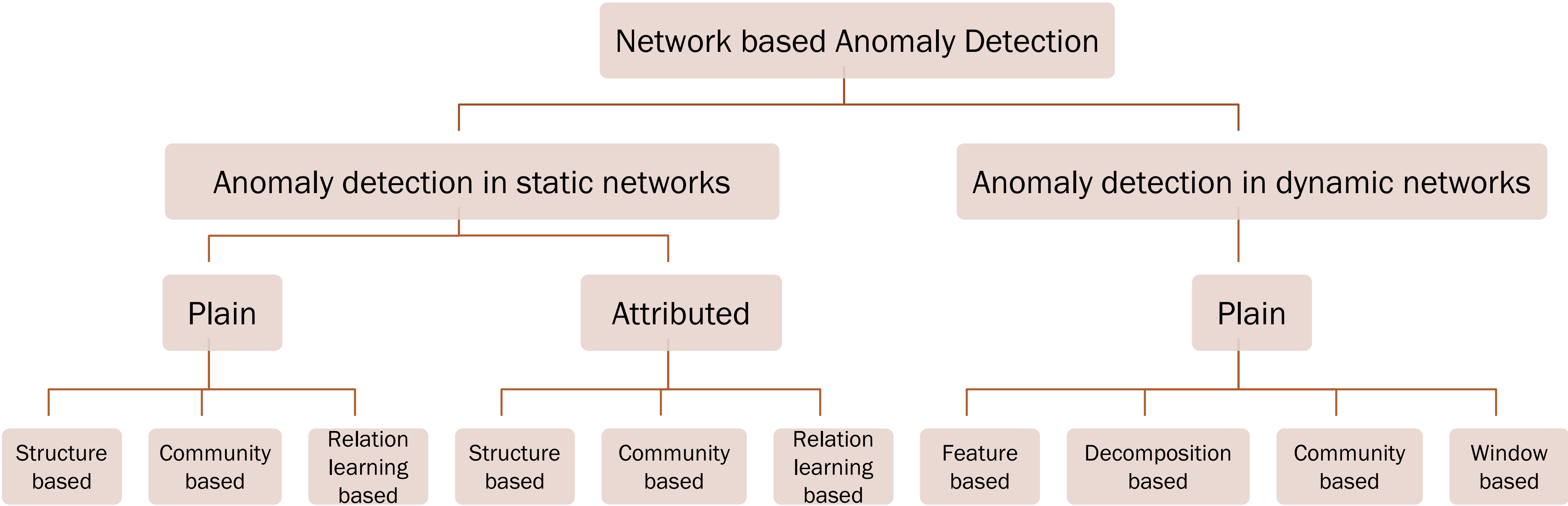
Class Imbalance: Anomalies are rare compared to normal data, resulting in a severe class imbalance.

Requirements for Anomaly Detection Algorithms: These algorithms should be able to discover new anomalies as networks evolve, handle various anomaly types, and provide explanations for flagged anomalies.

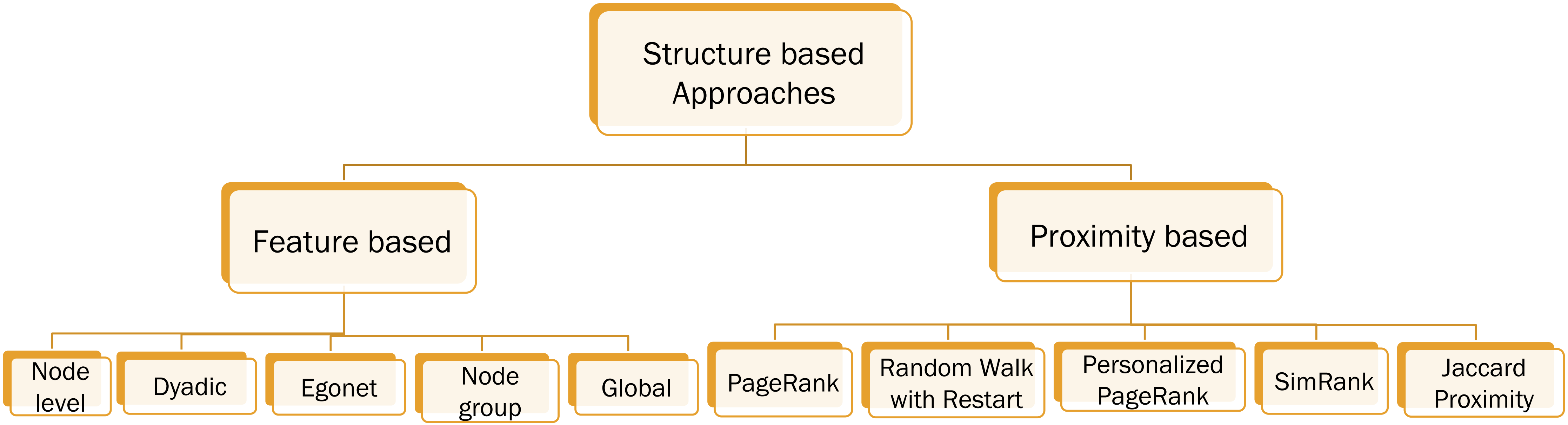
Network-based Anomaly Detection: Challenges

- Lack of labelled datasets wherein distinct labels are available for data points that are anomalous and non-anomalous
- Manual labelling is not a feasible option
  - real-world networks are huge and grow dynamically at a very high rate
  - Network entities have a lot of metadata information associated with them in the form of complex feature vectors
  - Human interpretation is very subjective
- Anomalies are rare – class imbalance in the underlying data is a severe issue
- Anomaly detection algorithms should
  - find novel anomalies in different evolving settings of the same dataset
  - be robust enough to understand different types of anomalies
  - explain why they are flagged as anomalous after extracting anomalies

Network-based Anomaly Detection: Taxonomy



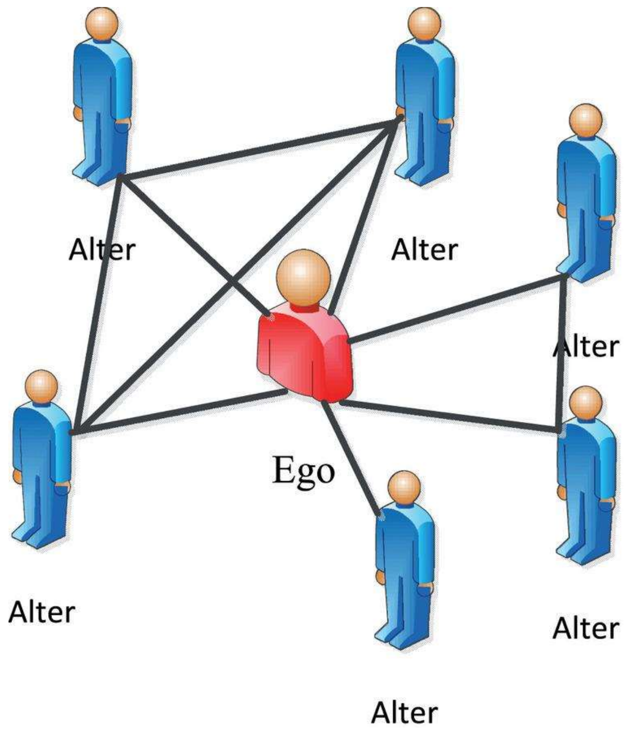
Anomaly Detection in Plain Static Networks: Structure-based Approaches



Anomaly Detection in Plain Static Networks: Network based Features

- Node level
  - Eigen vector
  - Closeness
  - Betweenness
  - Local Clustering coefficient
  - Degree Assortativity
- Global
  - Global Clustering Coefficient
  - Average Node Degree
  - Number of Connected Components
  - MST weight
- Dyadic
  - Reciprocity
  - Edge Betweenness
  - Common Neighbours
- Egonet
  - Number of triangles
  - Total Weight
  - Principal Eigen Value
- Node group
  - Density
  - Modularity
  - Conductance

Anomaly Detection in Plain Static Networks: ODDBALL



https://content.iospress.com/articles/journal-of-intelligent-and-fuzzy-systems/ifs190320

what is an oddball technique?

- One of the state-of-the-art feature based approaches to detect anomalies in a plain static network

- Proposed by Akoglu et al. in 2010 who proposed oddball technique... and when?

why does it use .... egonet base features = to focus the subnetwork je mara ego ae generate karyu 6e with its own immediate neighbors?

what does it use ?

- Extracts egonet-based features to focus on the subnetwork induced by the immediate neighbors of a node

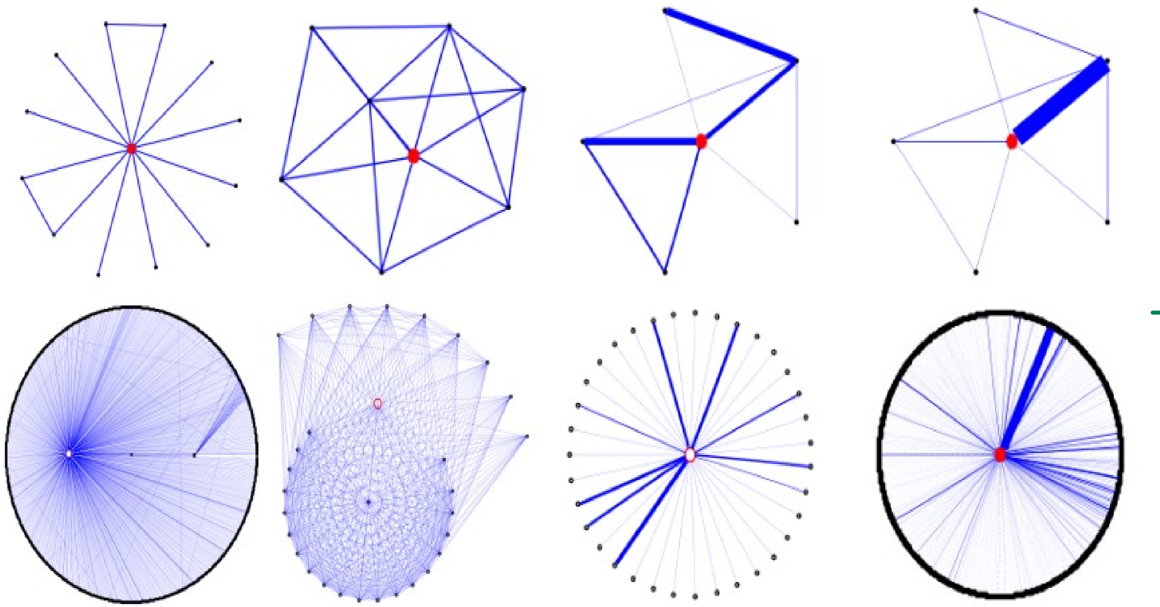
Egonet features :
1.Number of triangles 2.Total Weight 3.Principal Eigen Value

- Given a node (referred to as ego), its egonet is defined through a subnetwork induced by its one-hop neighbors (referred to as alters) or its immediate neighbors and the node itself

kaya hope neighbors ?
su kehvay 6e aemne

subgraph na andar kon-kon avse = alters + node itself.

Anomaly Detection in Plain Static Networks: Anomalies in ODDBALL



(a) Near-star (b) Near-clique (c) Heavy vicinity (d) Dominant edge

Types of ego :

- Near-cliques: egos whose alters are highly connected amongst each other
- Near-stars: egos whose alters are sparsely connected amongst each other

- Heavy vicinities: an edge with abnormally high weight with respect to the number of edges in its egonet aa vyakti na jode to kaik vadhare j bane 6e.
- Dominant heavy links: the ego node contacts one of its alters too frequently

tinu mama hemal bhai ne rojno call kare.

Anomaly Detection in Plain Static Networks: Features in ODDBALL

- Selected network Features:
  - N_i: degree of the ego node i
  - E_i: number of edges in the egonet of node i ketla number of edges 6e in the egonet of the given node i
  - W_i: total weight of the egonet (edges in the egonet) for node i
  - $\lambda_{w,i}$: principal eigenvalue of the weighted adjacency matrix of the egonet of node i

- Common advantages of these features:
  - efficient in terms of computation
  - provide suitable patterns describing normal behavior of a neighborhood structure

Anomaly Detection in Plain Static Networks: Features in ODDBALL

- Normal behaviour of nodes in ODDBALL:
  - Egonet Density Power Law:

$$E_i \propto N_i^{\alpha};\ \ 1 \leq \alpha \leq 2$$

  - Egonet Weight Power Law:

$$W_i \propto E_i^{\beta};\ \ \beta \geq 1$$

dwe

  - Egonet λ_w Power Law:

$$\lambda_{w,i} \propto W_i^{\gamma};\ \ 0.5 \leq \gamma \leq 1$$

Anomaly Detection in Plain Static Networks: Proximity-based Approaches

- PageRank
  - Proposed by Brin and Page in 1998
  - Provides importance of a node based on the importance of its neighbors
  - an **extremely high rank** (importance) for a node could be a **sign of an anomaly**
- Random Walk with Restart
  - A random walk based algorithm
  - At each step, the random walker has a small probability of restarting the walk from the source
  - Helps in exploring the neighborhood of the source nodes
- Personalized PageRank
  - similar to Random Walk with Restart
  - each node is associated with a chance of getting "teleported" to one of the nodes belonging to the given "teleportation set"

Anomaly Detection in Plain Static Networks: Proximity-based Approaches

- SimRank
  - a measure of **similarity between two nodes** in a network
  - Two nodes are similar if they are "referenced" by similar nodes
- Jaccard proximity
  - ratio of number of common neighbors between two nodes to the number of nodes in the union of neighbor sets of the two nodes
  - a measure of closeness or proximity
  - denotes the likelihood of an edge existing between the two nodes

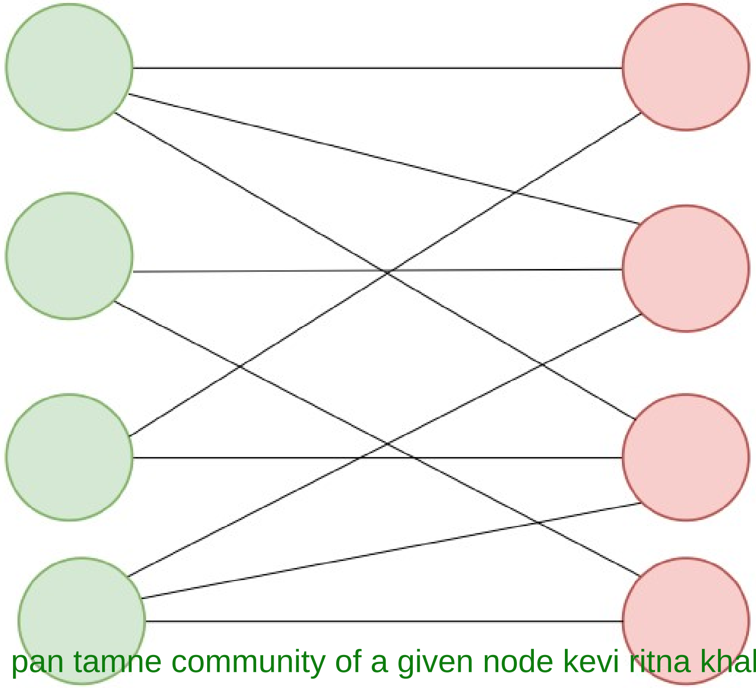
Anomaly Detection in Plain Static Networks: Community-based Approaches

what is this approach based on?

- Based on detecting communities to spot anomalous nodes and/or edges
- Entities (nodes and/or edges) that have a large number of cross-community relations
- Two state-of-the-art approaches
  - Anomaly detection in bipartite networks (Sun et al. 2005)
  - AutoPart (Chakrabarti 2004)

Anomaly Detection in Plain Static Bipartite Networks

What is a bi-partitie network?



- Networks that have two sets of nodes with no internal connections among the set members

Example of bi-partite network.

- A publication network can be modelled as a bipartite network

- Two partitions of nodes
  - Authors
  - papers written by the authors

what is our objective ?

- Objective: To find unusual/anomalous papers written by authors belonging to different research communities

atrangi or naveen papers that belong to the different reserch communities.

- Fundamental research questions:
  - How to detect the community of a given node?
  - How to quantify a measure to assess the level at which a node should be considered as a bridge node?

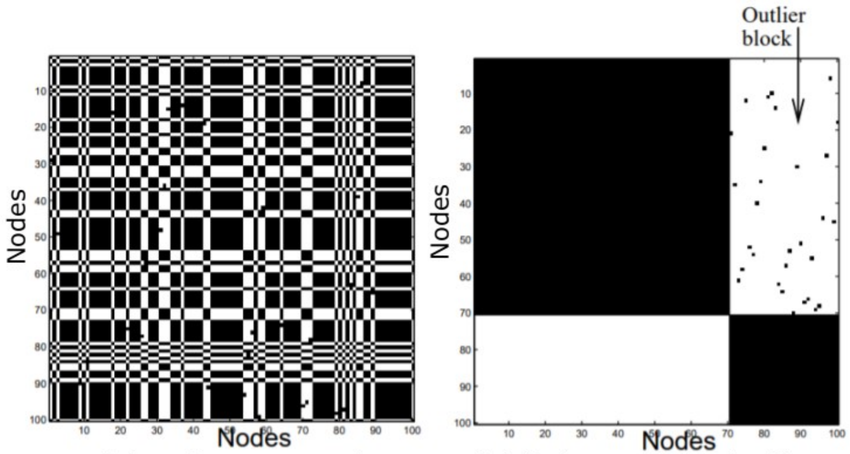
pan tamne community of a given node kevi ritna khabar padse?

kevi ritna ae node ni membership function ne value ne count karso for it to belong as a bridge node

Anomaly Detection in Plain Static Bipartite Networks

- To detect the neighborhood of a given node:
  - Use of random walk with restart based Personalized PageRank (PPR) what is used to detect the neighborhood of the given node ?
  - Algorithm computes the PPR scores for all the nodes in the network what does the ppr do to detect the neighborhood of a given node?
  - nodes with the highest PPR scores form the neighborhood of the given node
- To detect bridge nodes in the network
  - Based on a normality score for the node what we use to detect the bridge node
  - Obtained by averaging the pairwise PPR scores among all the neighbors of the node how to obtain the normality score?
  - Nodes with **lower normality score constitute bridge nodes** how does the normality will tell us which are the bridge nodes ?
- Random walker visits bridge nodes less frequently than community nodes → lower normality score for bridge nodes why do we have lower normality score for the bridge nodes rather than the community nodes ?

Anomaly Detection in Plain Static Networks: AutoPart



(a) Outlier-test graph (b) Node groups and outliers

- ❑A parameter-free graph partitioning and outlier detection strategy what is used here ?
- ❑Proposed by Deepayan Chakrabarti in 2004 who proposed this theorem
- ❑**Group nodes** of the network **using information-theoretic principles.** how to group the nodes ?

what do we do here ?

- ❑**Rearrange the rows and columns** of the network adjacency matrix **forming dense blocks/clusters** of highly connected nodes exploiting the Minimum Description Language principle

the structures forming the dense blocks/clusters = highly connected nodes ...they exploit the minimum description languate principlple

- ❑Low-density blocks are flagged as outlier/anomalous blocks

low density blocks => are flagged as outlier /anamalous blocks.

which method uses the infrequent structures to find out the anomaly?

Anomaly Detection in Attributed Static Networks: Structure-based Approaches

what does attributed networks have .

how does the attributed networks store the additional information ?

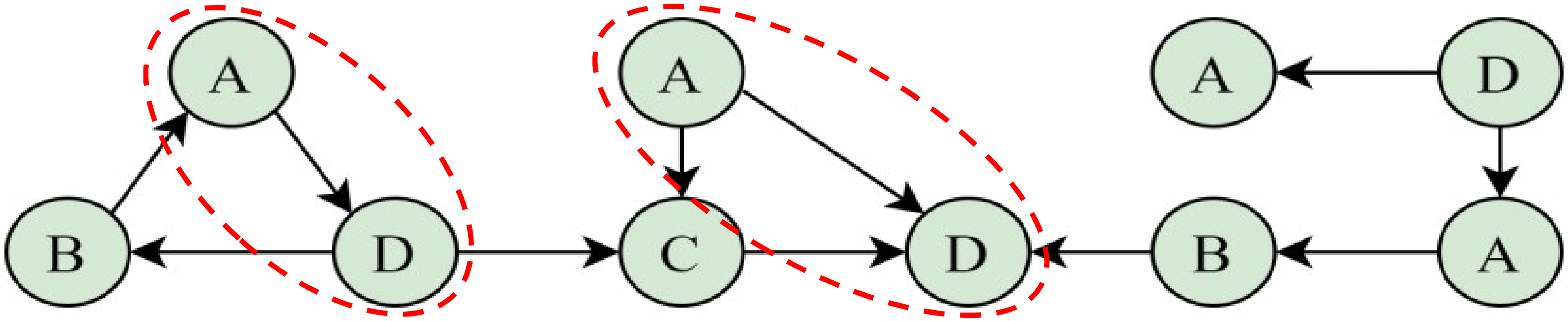
- Attributed networks have additional information in the form of node and/or edge attributes
- Basic network structure and meaningful insights extracted from node and/or edge attributes to detect unusual behavior

how do we do anamaly detection in attributed networks?

- Structure-based algorithms try to find infrequent substructures

how does structure based approaches do the anomaly detection.

- In example network, substructure (A ⟶ D) repeats twice in the network



what is Subdue?
what do we do in Subdue ?
Who proposed the method Subdue
When did they proposed the method Subdue?

Anomaly Detection in Attributed Static Networks: Structure-based Approaches

- Noble and Cook in 2003 proposed a method, Subdue, for finding network anomalies exploiting best substructures in the network

What is the fundamental method ?

- The fundamental principle of the Noble and Cook method include
  - To extract anomalous/unusual substructures in a network
  - To extract rare subnetworks from a set of subnetworks such that the nodes and/or edges contain categorical labels
- Best substructures are defined as those that occur frequently in the network

what are best substructures?

- These substructures help in compressing the network better

why are these substructures helpful?

Anomaly Detection in Attributed Static Networks: Compressing the Network

- Compressing a network structure refers to replacing substructures or subnetworks with a new node denoting the replaced substructure
- To evaluate the compression performance of a substructure preserving the network information quality, Minimum Description Length (MDL) is used
- Best substructures can be obtained by minimizing

$$F(S,G) = DL(G|S) + DL(S) \cdots\cdots\cdots (*)$$

$DL(S)$: description length of the substructure S

$DL(G|S)$: description length after compressing G using S

Anomaly Detection in Attributed Static Networks: Structure-based Approaches

- Extract anomalous/unusual substructures in a network:
  - Best substructures are those that occur frequently and generate lower values for Equation $(*)$
  - Anomalous substructures are those that produce relatively higher values for Equation $(*)$
  - an inverse variant of the MDL measure is an appropriate quantity to determine unusual substructures
  - The measure should take into account the size of the substructure
  - A heuristic measure for the purpose is:

$$F'(S, G) = size(S) \times instances(S, G) \cdots\cdots\cdots (**)$$

  $size(S)$: number of vertices in substructure S

  $instances(S, G)$: number of times S appears in G

- substructures with low values for Equation $(**)$ would be declared as anomalous

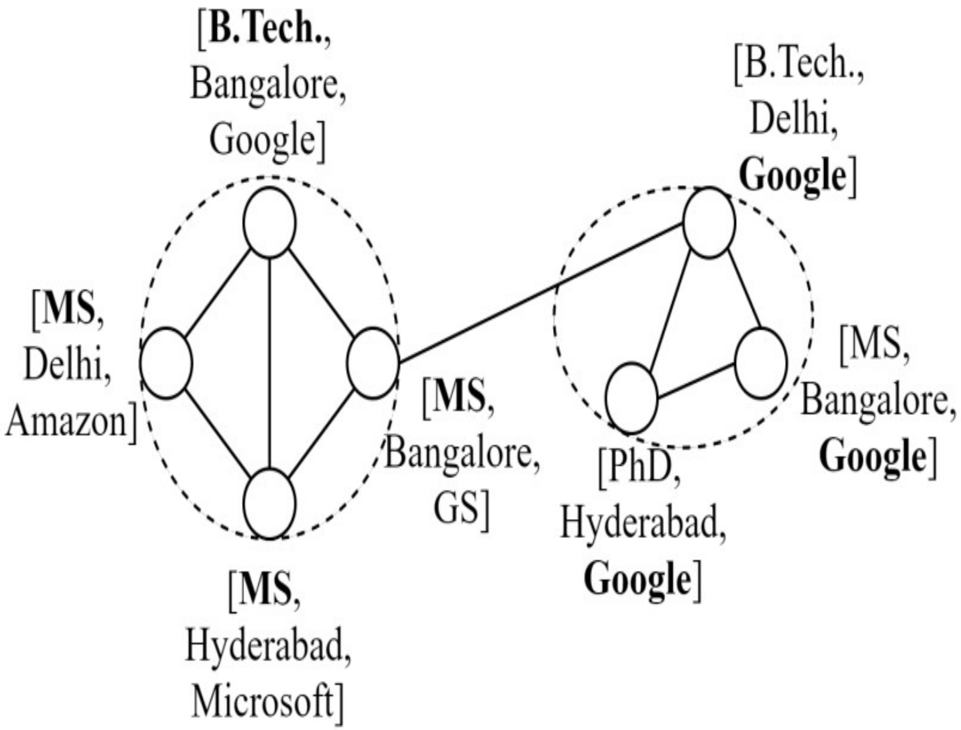
Anomaly Detection in Attributed Static Networks: Structure-based Approaches

- Extract rare subnetworks from a set of subnetworks:
  - use Subdue to find best substructures using Equation $(*)$
  - compress the subnetworks
  - Repeat above two steps as far as practicable
- Anomalous subnetworks would experience less compression in comparison to the normal ones
- Methodology proposed by Noble and Cook deals only with networks with categorical attributes
- Davis et al. proposedYagada for networks with numerical attributes in 2011
- Yagada is an adaptation of Subdue method in a different setting
- The method discretizes the numerical attributes such that the normal numerical attributes are all assigned the same categorical label and the anomalous attributes get their outlierness score

Anomaly Detection in Attributed Static Networks: Community-based Approaches

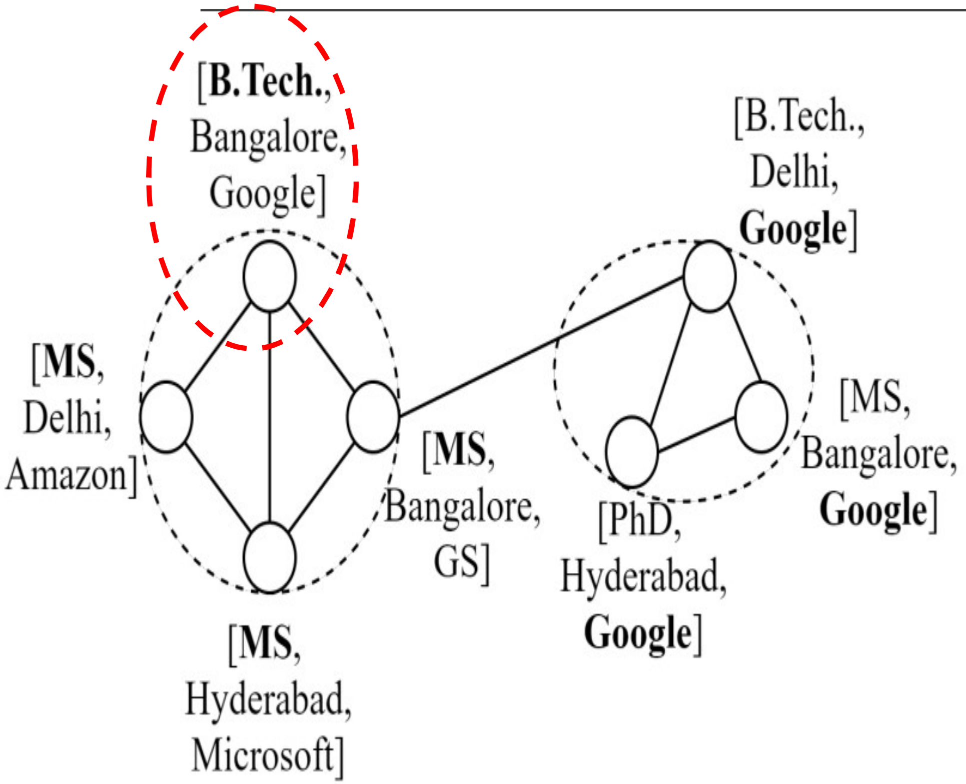
- Find community outliers that significantly differ from other community members based on their attribute values
- Two types of techniques:
  - Identify outliers along with detecting communities
  - Detect communities in attributed networks first and then extract anomalies
- FocusCO (Focused Clustering and Outlier Detection) extracts anomalous nodes while detecting focused clusters in a user-interactive manner

Community-based Approaches in Attributed Static Networks: *FocusCO*



- Focused Clusters:
  - Users provide a set of exemplar nodes that they perceive to be similar
  - Attribute weights are then inferred from these exemplar nodes
  - Attributes with large weights are termed as focus attributes
  - These attributes are used for focused clustering

Community-based Approaches in Attributed Static Networks: *FocusCO*



- Node attributes to example network
  - *highest degree attained*
  - *location of residence*
  - *place of work*

- Two focused clusters:
  - based on similar degree (left cluster)
  - based on similar place of work (right cluster)

- Node with attribute 'B.Tech.' is focused outlier in left cluster

Community-based Approaches in Attributed Static Networks: *FocusCO*

- Input:
  - $G = (V, E, F)$: Given attributed network with the set of node attributes F
  - C_{ex}: a set of exemplar nodes similar to those present in focus clusters preferred by a user u
- Output:
  - only those focus clusters C of G that correspond to the interests of u
- Given the inputs, *FocusCO* does the following:
  - Infers importance/weights (β_u) of all the attributes in F based on C_{ex}
  - Extracts focus clusters C in G using the attribute weights vector β_u
  - Detects focused outliers O such that they structurally belong to focus clusters C in G but differ significantly from other cluster members based on the focus attributes

FocusCO:
Infer Attribute Weight Vector β_u

1. Initialization: Similar pairs $P_S = \varphi$, Dissimilar pairs $P_D = \varphi$;
2. For $u \in C_{ex}$, $v \in C_{ex}$ do
   a. $P_S = P_S \cup (u, v)$
3. End
4. While $|P_D| \neq |F||P_S|$ do
   a. Sample $u, v \in V \setminus C_{ex}$
   b. $P_D = P_D \cup (u, v)$
5. End
6. Oversample P_S such that $|P_D| = |P_S|$
7. Get matrix A by solving the objective function /* f_v represents the feature vector for node v */

$$\min_A \sum_{(i,j)\in P_S} (f_i - f_j)^T A (f_i - f_j) - \gamma log\left(\sum_{(i,j)\in P_D} \sqrt{(f_i - f_j)^T A (f_i - f_j)}\right)$$

8. Return $\boldsymbol{\beta_u} = \boldsymbol{diag}(\boldsymbol{A})$

FocusCO: Extracts Focus Clusters C in G

- Find the candidate core sets using weights in β_u:
  - Re-weigh edges in E using feature similarity of end-nodes
  - Induce a subnetwork g of G with edges having comparatively higher weights
  - return Connected components in g
- Expand core sets by adding iteratively those non-member neighbours of nodes in core sets that brings the largest drop in weighted conductance:

$$\emptyset^{\omega}(C,G) = \frac{\sum_{(i,j)\in E,\, i\in C,\, j\in V\setminus C} \omega(i,j)}{\sum_{i\in C}\sum_{j,(i,j)\in E} \omega(i,j)}$$

- It is checked further if removing certain nodes in C can reduce the weighted conductance more

FocusCO:
Detect Focused Outliers

- Consider nodes during core expansion that are structurally best using unweighted conductance
- Call these nodes as *BSN*
- Nodes that are in *BSN* but not in *C* are detected as anomalies

Relational Learning for Anomaly Detection

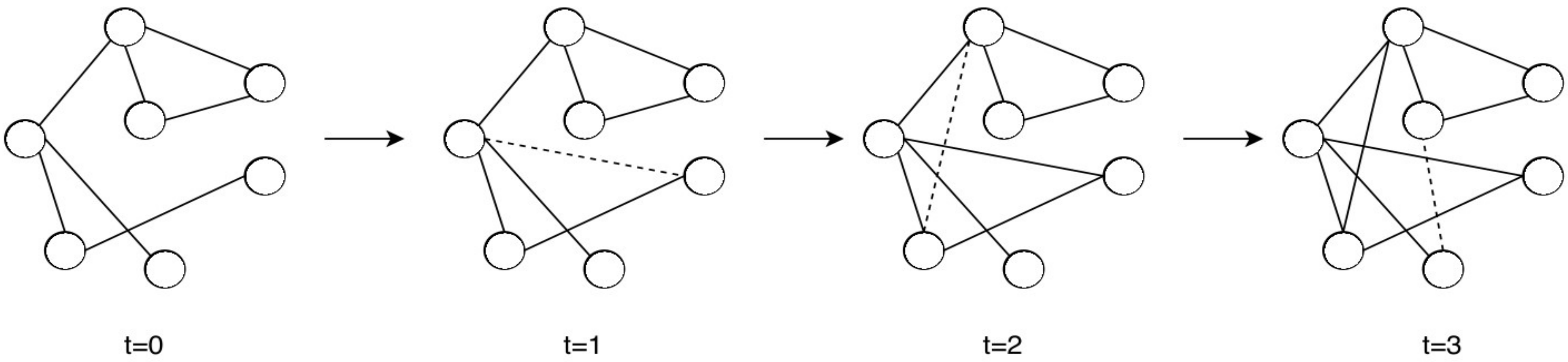
- Consist of network-based collective classification
- Working Principle: exploit the complex relationships between the data points to assign them into appropriate classes – normal and anomalous
- Can be formulated as a classification problem
- Usually complex in nature
- Example: In connection with the fraud detection problem
  - classify whether an online page is a spam page or not based on the keywords that appear on the page
  - Identify further whether it is a benign page or not

Relational Learning for Anomaly Detection

- Generally exploit the following inputs:
  - Unique attributes (features) of the nodes
  - Pre-labelled class of the node's neighbors
  - Unique attributes of the node's neighbors
- These relational inference algorithms can broadly be listed as:
  a. Loopy Belief Propagation
  b. Gibbs Sampling
  c. Iterative Classification Algorithm
  d. Weighted-vote relational network classifier
- Exact inference is known to be NP-hard in real-world networks
- Algorithms mentioned above are all approximate algorithms
- All of them are known to be fast, but convergence is not guaranteed

Anomaly Detection in Dynamic Networks

- Usually, the set of nodes remain constant and only edges change with time
- however, nodes can also change (get added or deleted) with time
- Often denoted by a sequence of static networks, which maintain a temporal order



Anomaly Detection in Dynamic Networks

- Given a temporal sequence of plain or attributed networks
  - First identify the timestamps at which a sudden change or event occurs, which forces in changing the network
  - Attribution: extract the top k-nodes, edges or parts of the network that contribute most to that change or event across two subsequent timestamps
- Anomaly detection algorithms in dynamic networks satisfy:
  - Scalability: able to process the updates in the networks over time
  - Sensitivity to changes in the structure or context of the network: should be sensitive to such changes like adding/removing nodes, edges or labels
  - Importance-of-change awareness: not all changes are important enough to track; must only track changes in important nodes

Anomaly Detection in Dynamic Networks: Categories

- Feature-based:
  - Highly similar networks usually share different network features
  - Typical features: degree distribution, clustering coefficient, diameter
- Decomposition-based:
  - Matrix decomposition of different temporal snapshots of the network
  - Based on eigenvalues, singular values or top eigenvectors
- Community-based:
  - Identify anomalies over evolving network clusters
- Window-based:
  - Initial temporal snapshots of the network are considered to be normal
  - later instances are studied against the initial snapshot to spot anomalies

Anomaly Detection in Dynamic Networks: Feature-based Approaches

- General idea for all feature-based methods:
  - Dynamic networks are nothing but temporal snapshots of a static network with some changes
  - Overall network properties like degree distribution, diameter, eigenvalues, etc. are similar across time
  - To extract a good summary of the network that can capture sensitive changes in the evolving network structure
  - Compare consecutive summaries using a chosen distance function
  - If the distance is more than a certain threshold set, the network is said to exhibit anomalous behavior in the corresponding timestamp
- The novelty of this category of methods lies in:
  - network summary
  - distance (or similarity) function
  - threshold

Anomaly Detection in Dynamic Networks: Generic Feature-based Approaches

1. Maximum Common Subnetwork (MCS) distance:
   a. maximum common subnetwork distance between the adjacency matrices of the consecutive snapshots, or
   b. maximum common subnetwork distance between the 2-hop matrices of the consecutive snapshots

2. Error correcting network matching distance:
   i. superimpose one snapshot of the network on another
   ii. edit the nodes, edges and weights to correct errors between the two snapshots
   iii. Count the number of operations required to transform one network to the other

3. Graph Edit Distance (GED):
   i. simplified version of error correcting network matching distance
   ii. only topological changes are allowed to edit
   iii. change in edge weights are disallowed

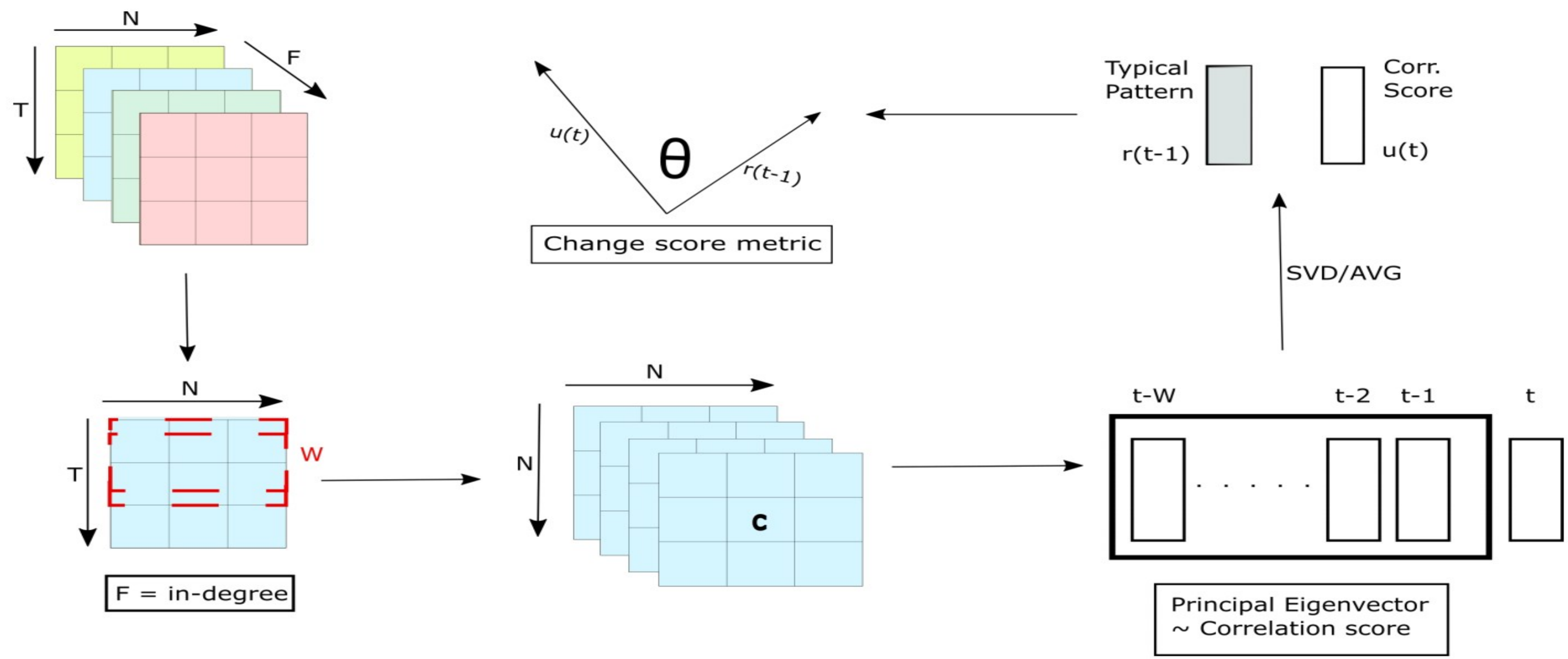
Anomaly Detection in Dynamic Networks: Generic Feature-based Approaches

4. Hamming distance:
   - count the number of differences between the adjacency matrices of two snapshots
5. Variations of edge-weight distances:
   - distance between two nodes is defined by the weight of the edge connecting two of them
6. Lambda-distance:
   - defined as the difference in the top-k eigenvalues of the respective adjacency, 2-hop or Laplacian matrices
7. Diameter distance:
   - defined as the difference in the network diameters

Anomaly Detection in Dynamic Networks: Akoglu and Faloutsos (2010)

- Underlying philosophy:
  - A node is anomalous if its current behavior deviates from its normal behavior based on past timestamps
  - At what point in time, do several nodes change their behaviors significantly?
  - Is it possible to characterize those nodes that change their behavior frequently?
- Each node is characterized on the basis of its egonet
- 12 features used to describe the node characteristic:
  - in-degree, out-degree, in-weight, out-weight, total degree
  - number of reciprocal neighbors, number of triangles
  - average in-weight, average out-weight, maximum in-weight, maximum out-weight
  - maximum weight ratio on reciprocated edges in the egonet

Akoglu and Faloutsos (2010): The Framework



Akoglu and Faloutsos (2010): The Method

1. The network is a $\mathrm{T} \times N \times F$ three-dimensional tensor
   1. N: number of nodes
   2. F: number of features
   3. T: number of timestamps
2. Extract a slice of the tensor for a feature; the shape of the data is $\mathrm{T} \times N \times 1$
3. Define W, a window size
4. For each pair of nodes, calculate the Pearson's correlation coefficient between their time-series vectors over the window of size W
5. Window slides by time-ticks, and correlation coefficients are calculated
6. For every window, we shall get a $N \times N$ matrix of correlation coefficients; $C = T - W$ such matrices

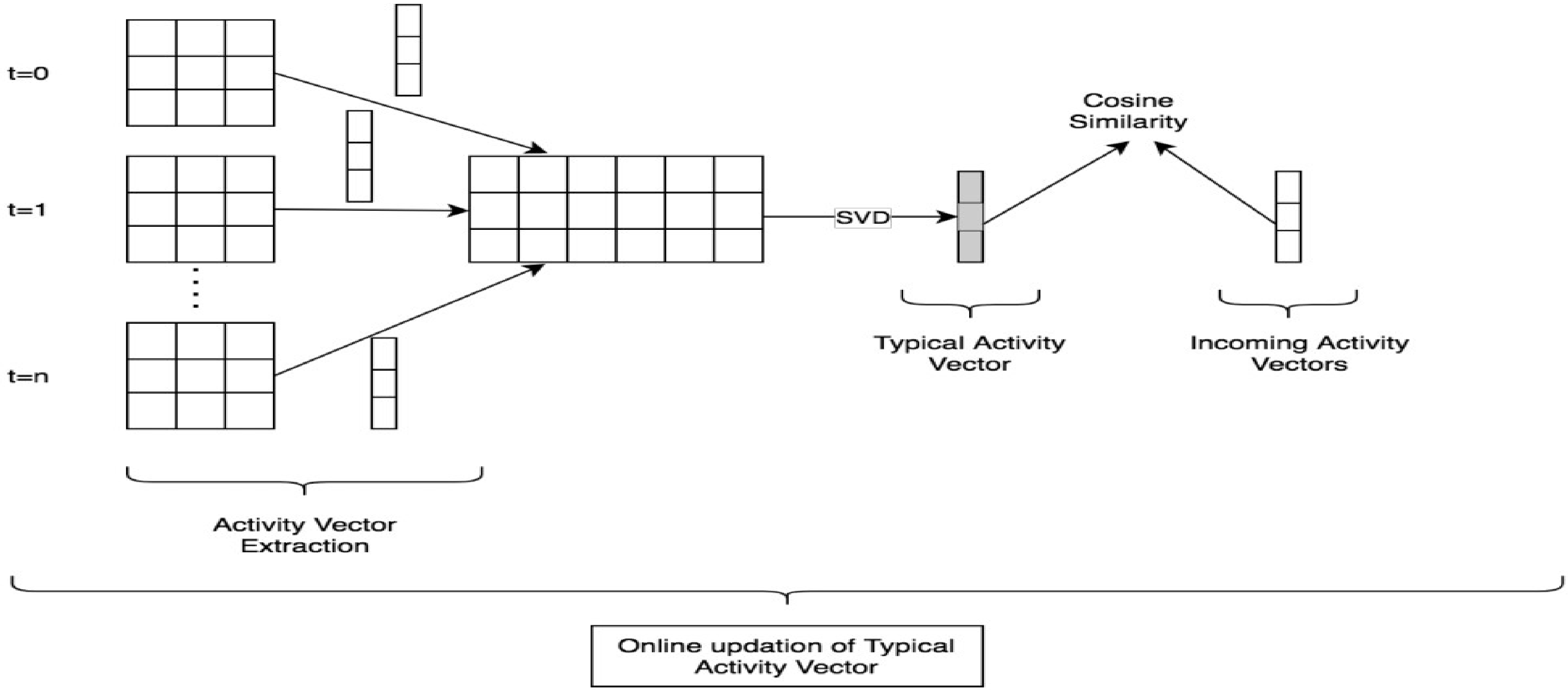
Akoglu and Faloutsos (2010): The Method

7. Extract the principal eigenvector for all of the C matrices of size $N \times N$
8. Compute typical-eigen-behavior$u(t)$, which is the principal eigenvector at time $t - r(t - 1)$
9. Note the change: $Z = (1 - u^T r)$; u is the eigenvector at time T and r is the normal eigen-behavior vector
    1. If $Z = 0$, no anomalous behaviour
    2. If $Z = 1$, anomalous situation

Anomaly Detection in Dynamic Networks: Decomposition-based Approaches

- Decomposition methods are a very powerful way to extract summaries of networks
- Decomposition-based methods require:
  - An indication of normal behavior
  - Similarity measure
- A simple approach for a decomposition-based method is proposed by IdÉ and Kashima in 2004
- Activity vectors: principal eigenvectors extracted from the adjacency matrix of each snapshot
- Get a typical activity vector by combining activity vectors of different time steps and applying SVD on the same
- Anomalies are estimated based on the distance between a typical vector and the incoming activity vector

Decomposition-based dynamic anomaly detection: IdÉ and Kashima (2004)



Anomaly Detection in Dynamic Networks: Community-based Approaches

- Focus on how the membership of nodes in different communities changes with the course of time
  - members of the same community in a network tend to behave similarly
  - flag nodes which lead to a significant change in the structural properties of the network, in the context of community membership
- Tasks of community detection and outlier detection occur simultaneously
- Typical example: ECOutliers proposed by Gupta et al. in 2012
  - Usually, members of the same community tend to behave similarly with time.
  - If some nodes in the network change their behavior significantly from the average behavior of the community, these nodes are referred to as Evolutionary Community Outliers

Anomaly Detection in Dynamic Networks: *ECOutliers*

- Some notations:
  - M – a matrix
  - $M_{i,*}$ – i^{th} row of matrix M
  - $M_{*,j}$ – j^{th} column of matrix M
  - X_i – i^{th} temporal snapshot of the network
  - $a \cdot b$ – inner product of vectors a and b
- A sequence of snapshots of the network: $X_1, X_2, \cdots, X_n$
- K_i – number of communities in X_i
- $P \in [0,1]^{N \times K_1}$ and $Q \in [0,1]^{N \times K_2}$ are the community belongingness matrices for X_1 and X_2, such that

$$\sum_{i=1}^{K_1} P_{o,i} = 1 \qquad \sum_{j=1}^{K_2} Q_{o,j} = 1$$

Anomaly Detection in Dynamic Networks: *ECOutliers*

- A soft correspondence to match communities in a pair of snapshots:
  - A correspondence Matrix: $S \in [0,1]^{K_1 \times K_2}$ such that

$$\sum_{j=1}^{K_2} S_{o,j} = 1$$

  - To learn the optimal correspondence matrix so that it gives the best matching between the communities in the two snapshots
- To quantify the anomalous behavior of node-community pairs:
  - an outlierness matrix$A$ of dimension $N \times K_2$
  - $A_{o,j}$: outlier score for node o and community j in X_2
- An object-community pair $(o,\ j)$ is an ECOutlier if the change from $P_{o,i}$ to $Q_{o,j}$ is very different than the change in belongingness in the other nodes between communities X_i and X_j

ECOutliers: Joint Framework for Estimating *S* and *A*

ECOutlier optimization problem:

$$\min_{S,A} \sum_{o=1}^{N} \sum_{j=1}^{K_2} \log\left(\frac{1}{A_{o,j}}\right) \left(Q_{o,j} - P_{o,*}S_{*,j}\right)^2$$

$$s.t. \quad S_{i,j} \geq 0 \qquad \forall i = 1,2,\cdots,K_1;\ \forall j = 1,2,\cdots,K_2$$

$$\sum_{j=0}^{K_2} S_{i,j} = 1 \qquad \forall i = 1,2,\cdots,K_2$$

$$0 \leq A_{i,j} \leq 1 \qquad \forall i = 1,2,\cdots,K_1;\ \forall j = 1,2,\cdots,K_2$$

$$\sum_{i=1}^{N} \sum_{j=1}^{K_2} A_{i,j} \leq \mu$$

- ❑ μ is the estimated sum of outlierness in a snapshot; its final value will also be learnt as we estimate A

ECOutliers: Joint Framework for Estimating S and A

- Using the method of Lagrange Multipliers for solving the optimization problem:

$$\min_{S,A} \sum_{o=1}^{N}\sum_{j=1}^{K_2} \log\left(\frac{1}{A_{o,j}}\right)\left(Q_{o,j} - P_{o,*}S_{*,j}\right)^2 + \sum_{i=1}^{K_1}\beta_i\left(\sum_{j=0}^{K_2} S_{i,j} - 1\right) + \gamma\left(\sum_{i=1}^{N}\sum_{j=1}^{K_2} A_{i,j} - \mu\right)$$

$$s.t. \quad S_{i,j} \geq 0 \qquad \forall i = 1,2,\cdots,K_1;\ \forall j = 1,2,\cdots,K_2$$

$$0 \leq A_{i,j} \leq 1 \qquad \forall i = 1,2,\cdots,K_1;\ \forall j = 1,2,\cdots,K_2$$

- Applying partial derivative with respect to $A_{o,j}$ and set it to 0, and simplifying

$$A_{o,j} = \frac{\left(Q_{o,j} - P_{o,*}S_{*,j}\right)^2}{\gamma}\ ; \quad \gamma = \sum_{o=1}^{N}\sum_{j=1}^{K_2}\frac{\left(Q_{o,j} - P_{o,*}S_{i,j}\right)^2}{\mu}$$

- Combining

$$A_{o,j} = \frac{\left(Q_{o,j} - P_{o,*}S_{*,j}\right)^2 \mu}{\sum_{o'=1}^{N}\sum_{j'=1}^{K_2}\left(Q_{o',j'} - P_{o',*}S_{*,j'}\right)^2} \cdots\cdots\cdots(\$)$$

ECOutliers:
Joint Framework for Estimating S and A

- Applying partial derivative with respect to $S_{i,j}$ and set it to 0, and simplifying

$$\sum_{o'=1}^{N} \left[2\log\left(\frac{1}{A_{o',j}}\right) (Q_{o',j} - P_{o',*}S_{*,j})(-P_{o'i}) \right] + \beta_i = 0$$

- The above yields,

$$S_{i,j} = \frac{\sum_{o'=1}^{N} 2\log\left(\frac{1}{A_{o',j}}\right) P_{o',i} \left[Q_{o',j} - \sum_{\substack{k=1 \\ k\neq i}}^{K_1} P_{o',k} S_{k,j} \right] - \beta_i}{\sum_{o'=1}^{N} 2\log\left(\frac{1}{A_{o',j}}\right) P_{o',i}^2} \cdots\cdots\cdots (\$\$)$$

- These are the update rules for $A_{o,j}$ and $S_{i,j}$

ECOutliers: The Method to Derive S and A

1. Initialize μ to 1
2. Initialize $S_{i,j} \leftarrow \frac{1}{K_2} \forall i,j$
3. Initialize $A_{i,j} \leftarrow \frac{1}{NK_2} \forall i,j$
4. While *not converged* do
   i. Update A using equation ($)
   ii. Update S using equation ($$)
5. End
6. $\mu \leftarrow \frac{\sum_{o'=1}^{N} \sum_{j'=1}^{K_2} \left(Q_{o',j'} - P_{o',*} S_{*,j'}\right)^2}{\max_{o,j}\left(Q_{o,j}^2\right)}$
7. Repeat steps 2 to 6

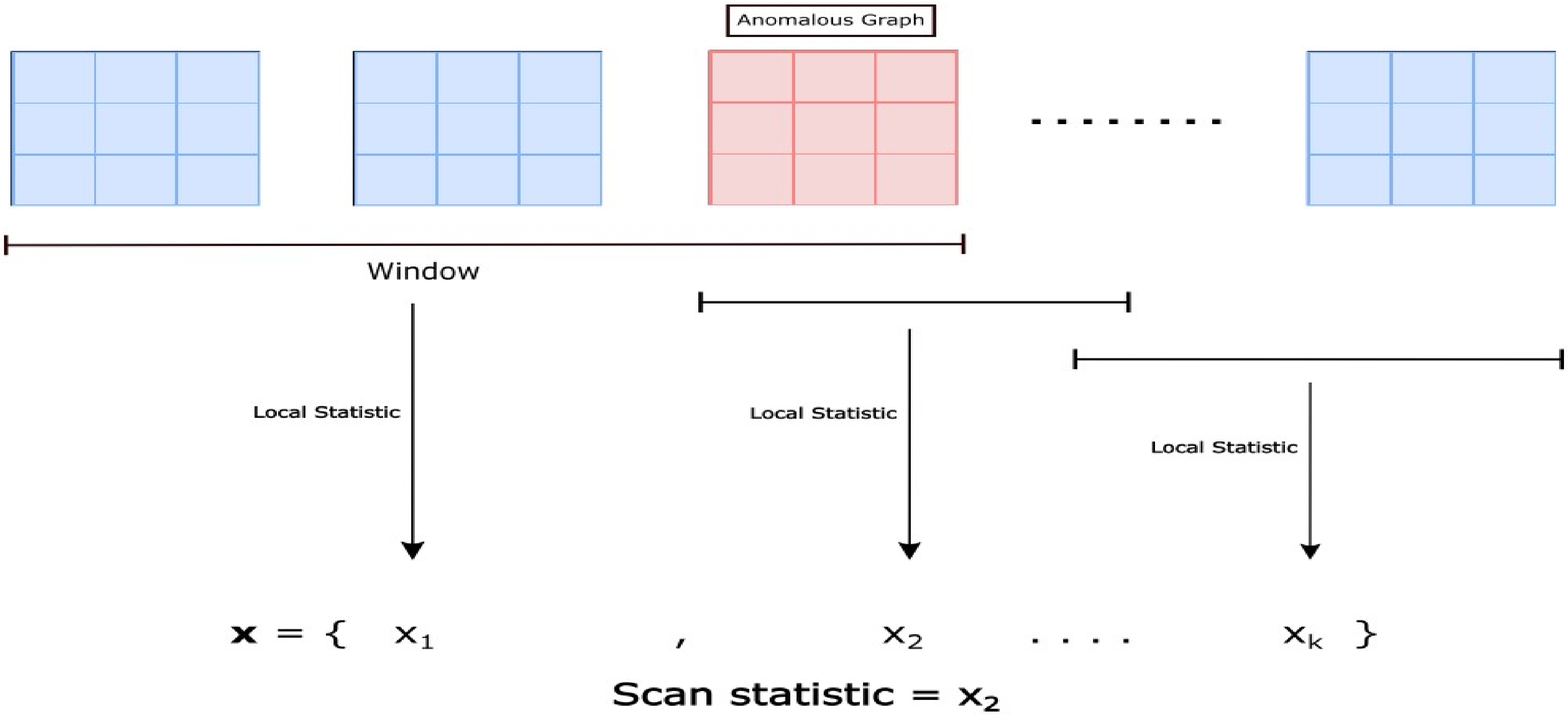
ECOutliers: Identifying the Outliers

- We now have outlier scores for every node with respect to every community in the network
- Multiple ways to proceed from here to detect outliers:
  - Entries of the outlier matrix A could be directly used for filtering outliers with respect to each community using some thresholding on the outlier scores
  - Aggregate the outlier scores for a node across all the communities
  - When enlisting the outliers in a given snapshot, one may consider the overall activity of the node during the snapshot

Anomaly Detection in Dynamic Networks: Window-based Approaches

- Instead of all the snapshots that have occurred previously, anomaly prediction is done based on snapshot history to a certain window
- Given a sequence of objects x, we can define a window as a subset of x, consisting of elements that occur sequentially in x.
- The number of elements in this subset is the window size
- The number of elements that are skipped between the starting points of two consecutive windows is called the hop length
- Example: a sequence of numbers $x = 1, 2, 3, 4, 5, 6, 7, 8, 9$
  - A window of window size 3 is $1, 2, 3$
  - If hop length is 2, next window would be $3, 4, 5$

Window-based Anomaly Detection in Dynamic Networks: Priebe et al. (2005)



Window-based Anomaly Detection in Dynamic Networks: Priebe et al. (2005)

- To spot anomalies by identifying snapshots that have unusually high connectivity, as compared to the previous time-steps
- A local statistic is a statistic that outputs a certain value for a selected window
- The maximum value of these local statistics is called scan statistic
- If the scan statistic exceeds a certain threshold, the corresponding window is determined to be anomalous

END