

Social Network Analysis
Prof. Tanmoy Chakraborty
Department of Computer Science and Engineering
Indraprastha Institute of Information Technology, Delhi

Chapter - 08
Lecture - 06

(Refer Slide Time: 00:27)

Dynamic Graphs: Decomposition based methods



- Detect temporal anomalies by resorting to matrix/tensor decomposition of the time-evolving graphs, and interpreting appropriately selected eigenvectors, eigenvalues or singular values.
- The methods can be divided in two categories based on the representation of the graphs: matrices vs. tensors.



So, now let us look at the decomposition based approaches ok. So, here the idea is more or less same. The idea is that we you know either construct a matrix or a tensor right and then we decompose it, right. The matrix decomposition matrix factorization we all know how to do this. So, we decompose it and basically the idea is that we again we try to generate a summary for every time stamp and we look at the differences between two summaries.

(Refer Slide Time: 00:52)

Dynamic Graphs: Decomposition based methods



- The method first extracts the principal eigenvector from the adjacency matrix of each graph – referred to as **activity vector**.
- Then, by applying SVD on the matrix that consists of the past activity vectors in a time window w , the typical activity vector is found.
- The similarity between the current and typical activity vectors is computed as the cosine of the angle between them.

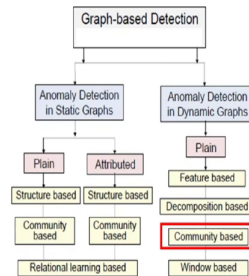


So, a method the method first extracts the principal eigenvector the same way from the adjacency matrix of the graph. In the previous case, we extracted the principal eigenvector from the from the similarity matrix right; the Pearson correlation matrix that we generated right, but here we extract the principle eigenvector from a from an adjacency matrix.

We call it as a activity vector, then we apply SVD on the matrix that consists of the past activity vectors in a time window w . And then the similarity between the current and the typical activity vectors in compute is computed as the cosine angle. Now, this is very similar to the one that we discussed earlier, ok.

(Refer Slide Time: 01:46)

Outline



So, let us now spend some time on another type of algorithm which detects anomalous entities based on the community structure ok.

(Refer Slide Time: 02:04)

Dynamic Graphs: Community based methods



- Monitor graph communities or clusters over time and report an event when there is structural or contextual change in any of them.
- Looks at the changes in two subsequent snapshots of the graph or all the snapshots of the graph.
- Detects communities and outlier simultaneously



So, right dynamic graphs community based methods. So, it monitor graph communities or clusters over time and reports an event when there is a structural or contextual change in any of them right. It looks at the changes in two consecutive snapshots of the graph or all the snapshots of the graph right. Detect communities and outliers simultaneously.

(Refer Slide Time: 02:38)

Dynamic Graphs: Community based methods

Evolutionary Community Outliers (ECOutliers) [Gupta et al., KDD'12]

- Most of the objects within a community follow similar *evolution trends* and their average defines the *evolution trend of the community*
- However, *evolutionary behavior* of certain objects is quite *different from the average evolutionary behavior* of its community.
- **Goal:** Detect such anomalous objects (ECOutliers) given a pair of snapshots



Let us look at one such algorithm and let us focus on this algorithm. This is called evolutionary community outlier algorithm, is also known as you know EC outliers. This was published in KDD 2012, ok. So, let us look at the intuition. So, most of the objects within a community follow similar evolution trends and their average defines the evolution trend of a community ok.

Let us assume that we have three communities at time stamp t one ok. And you focus on say let us say this community ok. And you see how this community behaves, how nodes move from this to this, this to this, right. So, generally if you know if this community is a normal community, its constituent nodes also behave similarly throughout the time stamps right. Say, this node will move to this node if other nodes also behave similarly.

They would also move to this community ok; so, but if there is any change in the evolutionary behavior of certain objects, then we say that this community is basically an outlier community. So, evolutionary behavior of certain objects is quite different from the average evolutionary behavior of its community, right. And the goal is to detect such anomalous objects right given a pair of snapshots. So, this is time stamp t_1 , time stamp t_2 , we look at every pair and we measure this abnormality.

(Refer Slide Time: 04:42)

Dynamic Graphs: Community based methods

Evolutionary Community Outliers (ECOOutliers) [Gupta et al., KDD'12]

Examples:

- A stockbroker who suddenly changes his portfolio and starts investing in another sector against his historical investments even when other similar stockbrokers continue to invest in the same old sector.
- Consider the two snapshots 1997 and 1998 when Soumen Chakrabarti changed his research area from "Parallel Systems" to "Data and Information Systems"
- In 2010, there were 13 papers in CIKM about personalization, while none of the other IR conferences like WWW, WSDM, SIGIR or ECIR focused so much on personalization in that year



Let us take an example, let us take a few examples. So, let us assume that a stockbroker right who suddenly changes his portfolio and starts investing in another sector against his historical investments even when other similar you know stockbrokers continue to invest in the same old sector. This is definitely an outlier behavior. Example 2: so, let us take the research career of say Professor Soumen Chakrabarti, who is currently a professor of computer science in IIT, Bombay.

So, till 1997, 1998, he used to publish papers in Parallel Systems parallel architecture Parallel Systems mostly systems conferences, but around 98, 99, he started publishing papers in you know data and information systems like conferences WWW, KDD, WSDM, etcetera. So, this is definitely an anomalous behavior because rest of his neighbors of the same community.

They still kept on publishing papers in system conferences whereas, he suddenly changed you know probably changed his research area and started publishing papers in other areas right. So, definitely this is a community based outlier with I mean with respect to that community.

In 2010, there are 13 papers in a CIKM conference. CIKM is a data mining conference and this 13 papers were about personalization right, where none of the other IR conferences like WWW, WSDM, SIGIR, they focused on so much on personalization.

So, if you look at data mining conferences you have CIKM as one node, KDD another node, WWW another node, WSDM another node. And most of these conferences do not focus on personalization; whereas, CIKM started focusing on personalization from 2000 from 2010 right. So, definitely with respect to this community CIKM is an outlier ok. So, these are called community based outliers in a dynamic graph, ok.

(Refer Slide Time: 07:24)

Dynamic Graphs: Community based methods

Evolutionary Community Outliers (ECOutliers) [Gupta et al., KDD'12]

Overall architecture

- Consider a snapshot series X_1, X_2, \dots, X_t
- We focus on a pair of snapshots (X_1, X_2)
- X_1 : P = matrix where P_{ij} denotes the probability of node i belonging to community j
- X_2 : Q = Similar matrix

Match the partitions of X_1 and X_2

Partitions of X_1

Partitions of X_2

Community com
- get split
- dis
- gaps etc.

Community mismatch happens due to outliers.

Hence outlier & community detection should be studied simultaneously.

So, let us look at the architecture now. So, we have a snapshot of graphs right X_1, X_2, X_3 dot dot dot X_t right. So, you can think of this as different snapshots right. And then you focus on a pair of snapshots right. So, this is one snapshot, this is another snapshot ok.

And you focus on a pair of snapshots say X_1 and X_2 right. P is a matrix corresponding to X_1 snapshot where each entry P_{ij} indicates the probability of node i belonging to community j right. So, this is a N cross K_1 matrix, N is a number of nodes K_1 is the number of communities in X_1 ok.


Similarly, we have another matrix Q this is N cross K_2 matrix. K_2 is the number of communities in at time stamp K_2 X_2 right. You see here if this is X_1 there are three communities K_1 is 3. If this is X_2 , then you have two communities right. And P_{ij} indicates the belongingness of node i to community j . How do you measure it? You can measure it in many ways.

For example, you can look at the number of edges of node i in community j right and so on. So, right we also make sure that each row entries of each row, sum of entries of each row should be 1 right, $P_{i,j}$. So, the total belongingness of a node to different communities is 1 and you have partial belongingness to different communities, right. Similarly, you have this one $Q_{i,j}$ 1 to K_2 is 1 ok. And this Q_1, Q_2 . This P and Q . You can easily measure from the algorithm right.

(Refer Slide Time: 10:32)

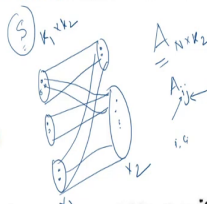
Dynamic Graphs: Community based methods

Evolutionary Community Outliers (ECOOutliers) [Gupta et al., KDD'12]




Notations

X_1	X_2	Snapshot	
K_1	K_2	# of Communities	
N	N	# of nodes	
$P \in [0,1]^{N \times K_1}$	$Q \in [0,1]^{N \times K_2}$	Belongingness matrix	
$\sum_{i=1}^{K_1} p_{ij} = 1$	$\sum_{j=1}^{K_2} q_{ij} = 1$		
$S^{K_1 \times K_2}$		Correspondence matrix	
$A^{N \times K_2}$		Outlier matrix	
	a_{ij}	outlierness score of i in community j	



$A^{N \times K_2}$
 A_{ij}

- An (Object, Community) pair is an "ECOOutlier" if change in p_{oi} to q_{oj} is quite different from the average change from community X_i to community X_j .



Let us look at some other notations. X_1, X_2 are snapshots K_1, K_2 are number of communities in X_1, X_2 respectively. Total number of nodes is fixed N , P and Q are or the belongingness matrices. This I already mentioned the sum of rho should be 1, then we introduce another matrix S . So, S is the correspondence matrix right. And this is K_1 cross K_2 number of communities in X_1 and number of communities in X_2 , right.

It basically says that say there are nodes here. Some of the nodes move to this community remaining nodes move to this community, right. So, this A is an association matrix between K_1 communities and K_2 communities ok. And this association matrix or correspondence matrix is something that we learn ok. Similarly, we have another matrix A which is N cross K_2 matrix.

This is called outlier matrix outlier course where A_{ij} indicates the propensity or the extent of outlierness of node i to community j ok, and what is the objective? So, an object community pair ok i, c_i for example, is an outlier if the change in p_{oi} to q_{oj} right; o is object from i th

community its moving to j th community in the next time stamp is quite different from the average change from community X_i to community X_j and our task would be to identify such outliers, ok.

(Refer Slide Time: 13:25)

Dynamic Graphs: Community based methods



Evolutionary Community Outliers (ECOutliers) [Gupta et al., KDD'12]

Integrated Framework

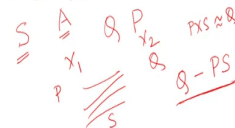
- Estimate $S^{k \times k}$ s.t. distance between Q and $P \times S$ minimizes.

- Such approach may be biased if we take into account "outliers"

↓
Need to ignore evolutionary outlier entries.

- Therefore, we incorporate A , outlierness score matrix.

- Develop integrated approach to compute S and A



So, now let us look at the algorithm right. So, it is called an integrated framework and what is the task here? The task is to estimate S right and A right. So, S is the correspondence matrix and A is the outlierness matrix that we estimate. And what are given to us? We have Q , we have P , right. Now, so, think about it.

So, at X_1 , you have this P matrix, you have some correspondence and you get X_2 and Q . So, ideally and this correspondence is S . So, ideally P times S should be same as Q right. So, this should be the ideal scenario. So, Q minus P times S would be the error, right.

(Refer Slide Time: 14:42)

Dynamic Graphs: Community based methods



Evolutionary Community Outliers (ECOutliers) [Gupta et al., KDD'12]

Integrated Framework

- M : Estimated sum of outlieriness in a snapshot

min $\sum_{i=1}^N \sum_{j=1}^{K_2} \log(a_{ij})$

Outlier weight for (i,j)

- $\log(\cdot)$ smooths weights, small range

- higher a_{ij} , lower weight \Rightarrow lower weight will be associated with (i,j) entry when performing community matching

Spurious error for community matching

Constraints

$-(S_{ij}) \geq 0 \quad \forall i=1 \dots K_1, \forall j=1 \dots K_2$

$-\sum_{j=1}^{K_2} S_{ij} = 1 \quad \forall i=1 \dots K_1$

$-\sum_{i=1}^N a_{ij} \leq M \quad \forall j=1 \dots K_2$

Lagrange Multiplier



So, look at here. So, the this is the function. We are minimizing the error right between Q and P times S right. We are minimizing this. And the problem here is that if we minimize right it would basically be biased towards outlier entities right. So, we need to add some quantity which you know reduces the bias of overall algorithms towards outlier entity. So, for that what we incorporate? We incorporate a log loss not a log loss as such, but a log weight right this is 1 by A . So, let us look at this one ok.

So, what is a_{ij} , a_{ij} is the outlier score of node i to community j ok. And this is the error. So, higher this outlier score this would be lower. So, this is basically weighted sum. This would contribute less towards this error. So, the node the nodes which are outliers those will contribute less towards this sum, right. And instead of taking direct weight we take the log weight right.

Again Q is known, P is known, S is unknown and A is unknown right. So, we minimize this objective function with respect to S and A . And what are the constraints? So, in the correspondence matrix S , the entries should be non-zero right. 0 is ok, but it should not be negative right. It should not be negative sorry it is 0 or greater than 0 , not non-zero, 0 or greater than 0 , but it should not be negative right for each for all pairs of K_1 and K_2 communities for a given i the sum should be 1 , right.

So, if we think of S , K_1 cross K_2 right. Each row the sum should be 1 , right. The entry in the outlieriness matrix, each entry should be positive, 0 or positive should not be negative.

And this is very important. The sum of all the entries of this A matrix which is N cross K 2, the sum of all entries should be less than mu, what is mu? Mu is an user defined constant right.

So, what I am saying is that it should not happen that you arbitrarily say that all the entities are outlier. You can say that if you do not incorporate this constraint, what would happen is that if I say everyone all the entries as outliers, right. So, all the entries would be outlier.

Therefore, the contribution should be less for all the entries and this should be the overall objective function should be minimum right. So, we say that this should not happen the total sum should be less than equals to mu ok. So, this is the objective function and these are the constraint.

So, this is the constraint objective function, constraint optimization function right. And how do we solve this problem? We see that there is equality and inequality constants right. How do you solve this? We use it using, we basically solve this using normal Lagrange multiplier.

So, those who do not know, what is Lagrange multiplier right please go back and check if you are familiar with SVM, Support Vector Machine you should definitely know what is Lagrange Multiplier. But those who do not know go back and check what is Lagrange multiplier. Lagrange multiplier is a way to move all the constant all these constraints to the objective function to make a single objective function to reduce the number of constraints right.

(Refer Slide Time: 19:40)

Dynamic Graphs: Community based methods

Evolutionary Community Outliers (ECOutliers) [Gupta et al., KDD'12]



Why M is needed

- Without M , one can simply mark all entries as outliers.
- Normal entries have small a_{oj}
- Outliers have large a_{oj}

+ S, A

Using Lagrangian Multipliers

$$\min_{S, A} \sum_{o=1}^n \sum_{j=1}^{K_o} \log \left(\frac{1}{a_{oj}} \right) (a_{oj} - \vec{b}_o \cdot \vec{S}_j)^2$$

$$+ \sum_{j=1}^{K_1} \beta_j \left[\sum_{j=1}^{K_o} S_{ij} - 1 \right] + \gamma \left[\sum_{o=1}^n \sum_{j=1}^{K_o} a_{oj} - M \right]$$

Subject to

$$S_{ij} \geq 0$$

$$1 \geq a_{ij} \geq 0$$



So, we see here that the this one right has been moved to a to the objective function here right and the last one has also been moved to the objective function. So, this was the old objective function. Now, these are two other constraints that have been added to the objective function right.

So, we now want to minimize this, how do we minimize it? We can minimize it with respect to S and A right. And simply we can use gradient descent type algorithms, minimization of gradient descent. So, we take the partial derivative of this with respect to S separately with respect to A , right.

(Refer Slide Time: 20:37)

Dynamic Graphs: Community based methods

Evolutionary Community Outliers (ECOutliers) [Gupta et al., KDD'12]



Taking the partial derivative with respect to a particular a_{ij} and setting it to 0, we obtain the following.

$$a_{ij} = \frac{(q_{ij} - p_{ij} \cdot s_{ij})^2 \mu}{\sum_{i=1}^N \sum_{j=1}^{K_2} (q_{ij} - p_{ij} \cdot s_{ij})^2}$$

Now, we will obtain the update rule for s_{ij} . Taking partial derivative of f with respect to s_{ij} , we obtain the following.

$$s_{ij} = \frac{\sum_{i=1}^N 2 \log \left(\frac{1}{a_{ij}} \right) p_{ij} \left[q_{ij} - \sum_{k=1, k \neq i}^{K_1} p_{ik} s_{kj} \right] - \beta_i}{\sum_{i=1}^N 2 \log \left(\frac{1}{a_{ij}} \right) p_{ij}^2}$$



So, this is the updation rule for A. If we take the partial derivative you see that this would be the updation rule for A, this should be the updation rule for S, right.

(Refer Slide Time: 20:50)

Dynamic Graphs: Community based methods

Evolutionary Community Outliers (ECOutliers) [Gupta et al., KDD'12]



Algorithm 1 OneStage μ Outlier Detection Algorithm

Input: P, Q

Output: Estimates of S and A

1: Initialize μ to 1

2: Initialize all $a_{ij} \leftarrow 1$ and all $s_{ij} \leftarrow \frac{1}{NK_2}$.

3: while NOT converged do

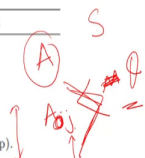
4: Update A using Eq. 10 (Outlier Detection Step).

5: Update S using Eq. 12 (Community Matching Step).

6: end while

7: $\mu \leftarrow \frac{\sum_{i=1}^N \sum_{j=1}^{K_2} (q_{ij} - p_{ij} \cdot s_{ij})^2}{\max_{i,j} (q_{ij})}$

8: Repeat Steps 2 to 6.



So, let us look at the algorithm now, the entire algorithm. So, we start off by assigning μ as 1 right by assigning all the entities of S as 1 by K_2 , because K_2 , K_2 is the maximum I mean K_2 is the sum of all the entries of S of S , right. We initialize a by 1 by NK_2 ; NK_2 is a sum of entries of all the of sum of entries of A right. We assign it this is the initialization then we update A and S based on these and these equations, right.

And we keep on doing this thing until and unless we converge right. There is no change of S matrix and A matrix. And then we update mu right. How do we update mu? Mu is now the fraction of the actual error that is still there right. You see here $Q - P \times S$. This is the error this is the sum square error sum square error. And this is normalized by the maximum number present in Q.

So, this is the normalized error. Now, this is our mu I repeat the same process again right. So, at the end of the day what I will have I will have A and I will have S right. A is the outlierness matrix which we actually need right. So, remember A_{ij} or say A_{oj} indicates the outlierness score of object o with respect to community j. Now, if this is less than mu right or say not exactly mu.

If it is less than certain threshold theta that you define right or say greater than because this outlier score. If the outlier score is greater than theta for example, then you say that object o is outlier with respect to community j ok. So, this is about EC outliers you can read this beautiful paper. This is a very interesting paper published in KDD and we have essentially exhausted.

We have discussed you know different types of algorithms for outlier detection we have mostly exhausted the taxonomy that we started with, right. So, this brings us to the end of this chapter. So, what we have learnt so far? We have learnt how to define an outlier, outlier is an illdefined problem right depending on certain application we define outliers.

Then, we look at static we looked at static graph and different types of algorithms for static graph feature based non feature based, community based. We look at dynamic graph feature based, decomposition based, community based and so on right. And then we have seen how we can we use outliers outlier detection methods for different applications fraud detection, cyber crime, right.

Anomaly detection outlier, fake news detection, fake account detection, credit card fraud detection and so on and so forth ok. So, this is in the end of this chapter. So, we will in the next chapter we will discuss we will start discussing on graph representation learning. We will first start with some basics of neural networks, deep learning and neural networks, and then we move to the graphic representation learning chapters, ok.

Thank you.