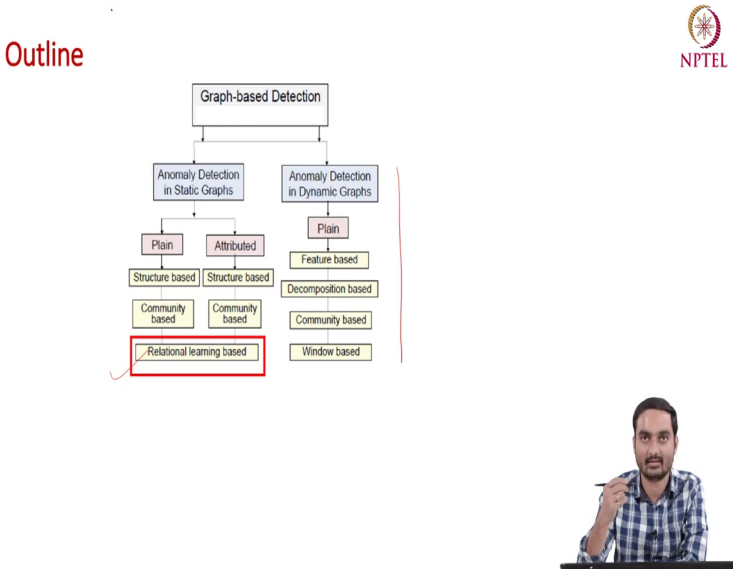
Social Network Analysis
Prof. Tanmoy Chakraborty
Department of Computer Science and Engineering
Indraprastha Institute of Information Technology, Delhi

Chapter - 08
Lecture - 05

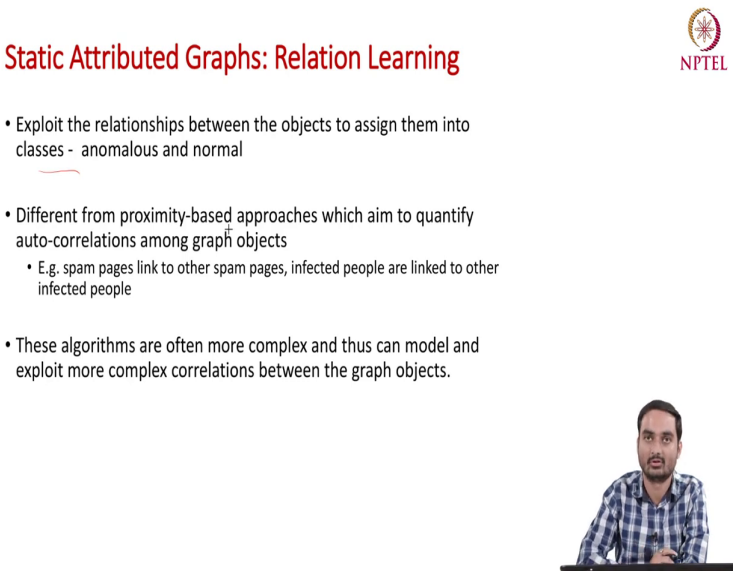
Alright. So, let us proceed with the taxonomy and algorithms, that we have been discussing.

(Refer Slide Time: 00:29)



So, let us look at the some of the algorithms that we use right for anomaly detection, with relational learning methods, right.

(Refer Slide Time: 00:38)

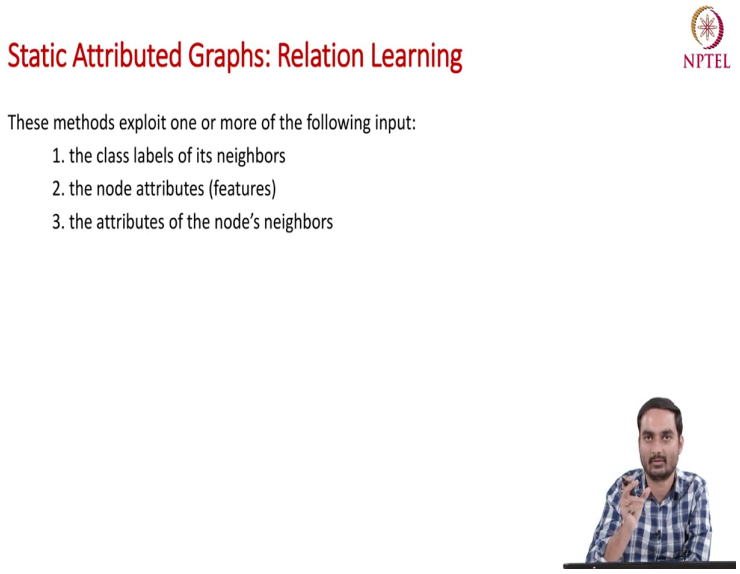


So, if you look at the relation learning kind of methods, they mostly take into account the relation between objects, right. For example, if a if there are two spam pages and they are linked, right. We will also look at the fact that nodes are spam pages and links this particular link is found between two spam pages right.

Similarly, we particularly take into account the fact that there is a link from a spam page to a non-spam page. So, it basically exploit the relationships between the objects to assign them into classes anomalous and non anomalous normal. It is different from the proximity based approaches that we have discussed so far.

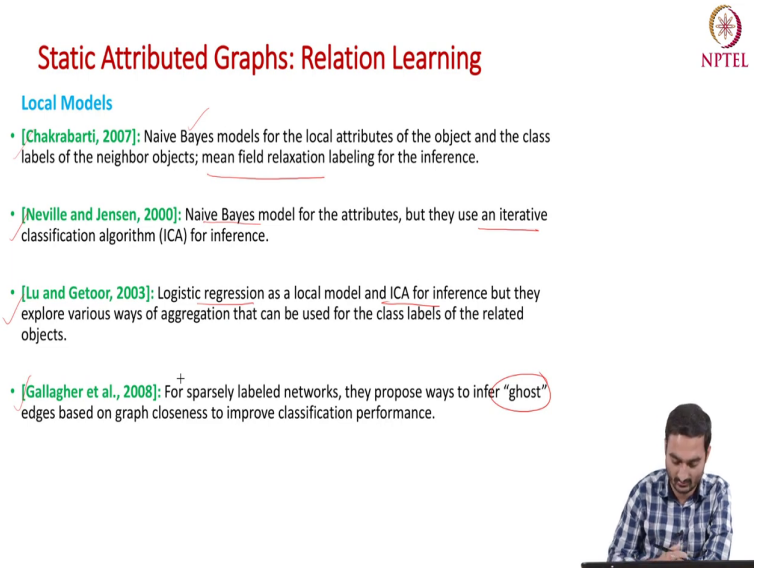
The proximity based approaches aim to quantify the autocorrelation between two objects, but here we aim to quantify the relations between two objects. So, these algorithms are even more complex, because apart from the topology we also look at the you know meta properties of edges and nodes.

(Refer Slide Time: 01:41)



So, if you look at the methods, these methods mainly exploit you know the classes of different nodes, if it is an a supervised algorithm then it takes into account the class information during training. It also uses the node attributes, node features and attributes of node's neighbors, so right. So, I am not going into the details of any of these algorithms, but I will try to give you a brief of what they basically do.

(Refer Slide Time: 02:13)



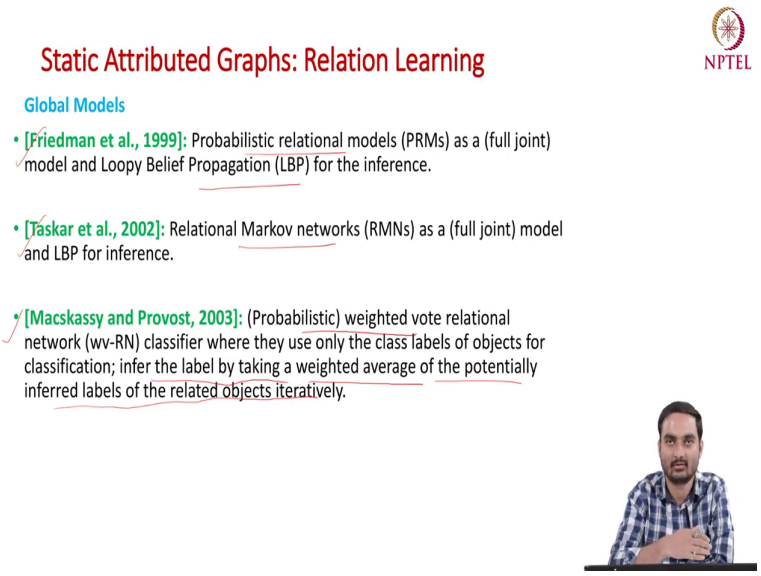
For example the algorithm proposed by Chakrabarti in 2007, it basically considers a Naive Bayes algorithm with some local attributes of objects and class labels and the I mean the

attributes of their neighbors as well. And they try to do some sort of mean field approximation labeling. For that there is another algorithm which simply uses Naive Bayes kind of methods, but they actually use it in an iterative manner, using some iterative classification method ICA.

The similar such method has also been proposed in 2003 by Lu and Getoor, who basically looked at the logistic regression and the similar kind of ICA method you know to identify classes of various objects. There is another algorithm which basically consider something called the ghost edges.

The ghost edges are those edges which you know you artificially create, based on either the similarity between objects or some other properties. And then try to measure the closeness as well as the as well as take into account the you know attributes of nodes and edges together ok, for the classification.

(Refer Slide Time: 03:38)



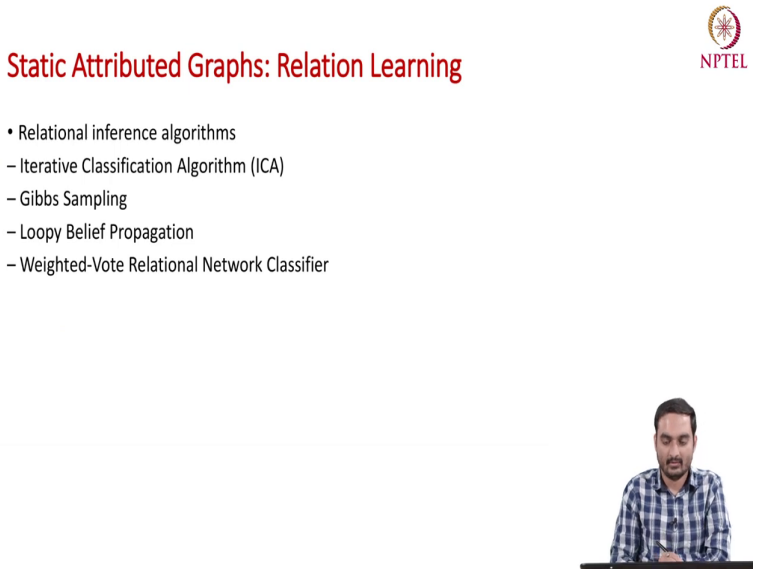
Similarly, there was another algorithm which considered some sort of; some sort of probably probabilistic methods as well as you know this belief propagation, label propagation test methods. For inference there is another method which considers relational Markov network right and again this belief propagation method, for classification.

Another method considers again probabilistic weighted some sort of weighted vote relational network for classification and they infer the label by taking a weighted average of the

potential inferred labels of the related objects iteratively. It is also an iterative algorithm, since this is based on belief propagation it also looks at the neighbors and aggregate the information, aggregates the information from the neighbors to classify the label of a to classify the node the given node, ok.

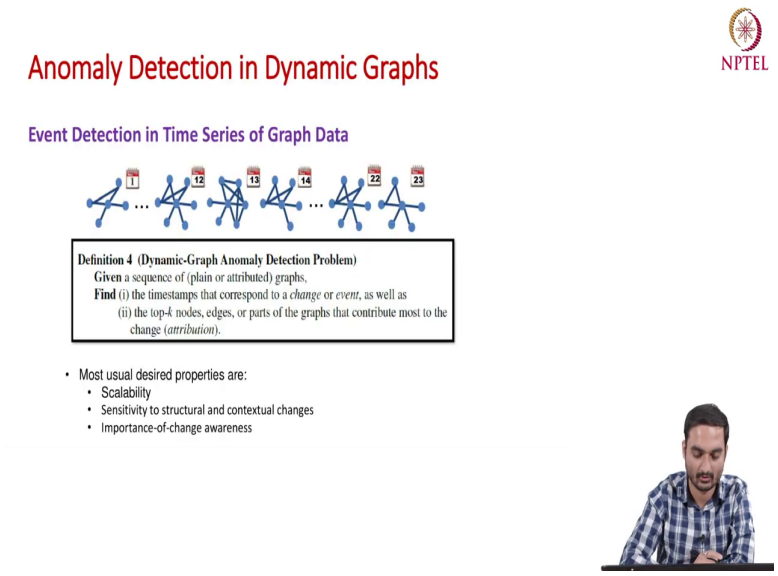
So, I am not going into the details of this algorithm because they are quite generic, these methods can also be used for non-graph based anomaly detection methods, but these are useful ok.

(Refer Slide Time: 04:50)



So, if you look at in general, people took into account the iterative classification algorithms, that is; this was very famous. They also took some sort of sampling algorithms like Gibbs sampling, because of this imbalance classification problem, the belief propagation method mostly has been used and some sort of weighted voting approach was used for final classification, ok. Let us move to the next part; so, anomaly detection in the dynamic graph ok.

(Refer Slide Time: 05:27)



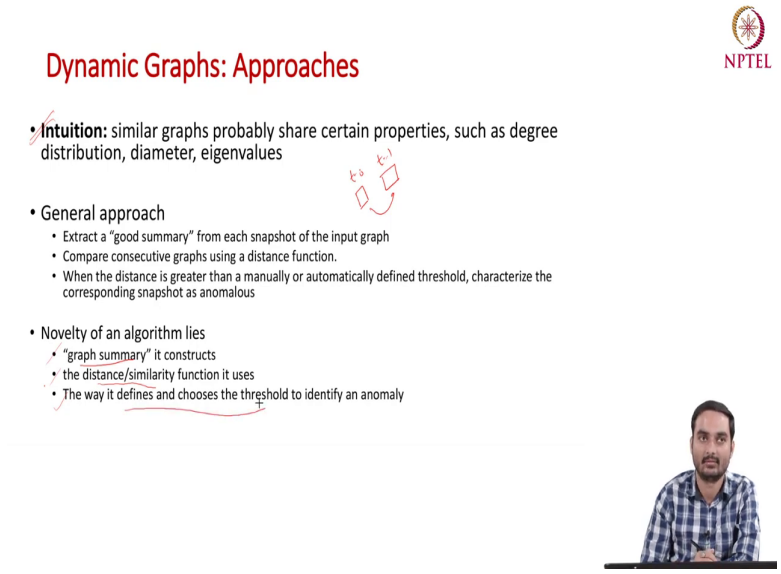
So, what is the problem statement here? The problem statement here is that you have a sequence of plane graph or attributed graphs right at time t 0 say t equals to 0, t equals to 1, t equals to 2, t equals to 3, t equals to 4 right, t equals to 5. And then what is your task? Your task is to come up with a timestamp right, where the change of topological structure of the graph between two consecutive timestamps is significantly different, significantly is significant, right.

So, the timestamps that corresponds to a change or event as well as the top k nodes and edges or a part of a graph that contribute mostly to the change ok. And what are the desirable properties? The desirable properties include the scalability right of the algorithm, the sensitivity to structural and contextual changes and importance of change awareness. So, it should be scalable, it should be fast enough right it should be sensitive to you know small changes for example.

I mean say if there is small change between two subsequent consecutive timestamps, we may not consider this as anomalous behavior, but if there is a significant change between two consecutive timestamps. We considered it as a you know as a kind of an anomalous behavior.

And then importance of change awareness it is very important. We need to understand whether this change is really important ok. So, if there is a, there is a small change or insignificant change we should not consider this as an anomalous behavior right.

(Refer Slide Time: 07:32)



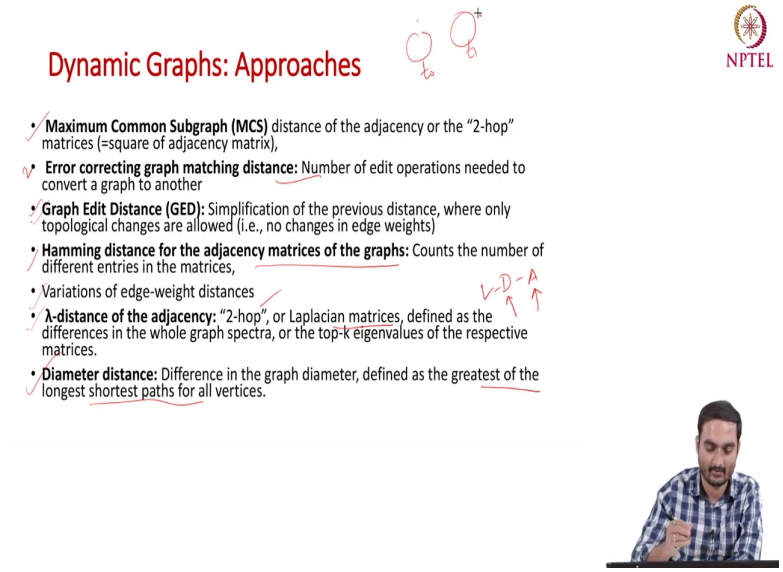
If you look at the feature based algorithms, people what people generally do? People generally you know. So, this is the intuition similar graph graphs probably share certain properties, such as the degree distribution diameter and eigen values. And what is the generic approach?

The generic approach is that you try to extract a good summary right of every timestamp graph right, you have timestamp t 0, you have a summary of the graph, timestamp t 1, you have the summary of the graph and then you compare subsequent consecutive graphs using some sort of distance measure.

The distance can be edit distance or say distance can be any graph based distance and when the distance is greater than some sort of manually set up threshold right, then we say that ok this is basically a change right, an event. So, all the algorithms which basically you know take into account the dynamic network graph dynamic properties of the graph. So, the novelties are in the construction of the graph summary right.

The distance measurement matrix formulation and the way it defines and chooses the threshold to identify the anomaly, right. So, we will see that the algorithms basically contribute to summarize the graph, contribute to finding out the similarity or dissimilarity measurement for measuring the distance between two summaries. And then it also chooses a threshold and based on that it basically says that if this is above the threshold then, this is an anomaly otherwise not ok.

(Refer Slide Time: 09:27)



So, if you look at the graph distance measures, people use matrix like maximum common sub graph, it is basically distance of two adjacency matrix or a second hop matrix right, of two consecutive timestamp graphs. Either correcting graph matching distance, number of edit operations needed to convert a graph to another.

This is basically an edit distance kind of approach and then graph edit distance a simplification of this previous one, where only topological changes are allowed right. No change is the edge weight, right. So, in this in metric two it also considers the weight of the edge, but this metric does not consider the weight of an edge, but only consider the topological changes.

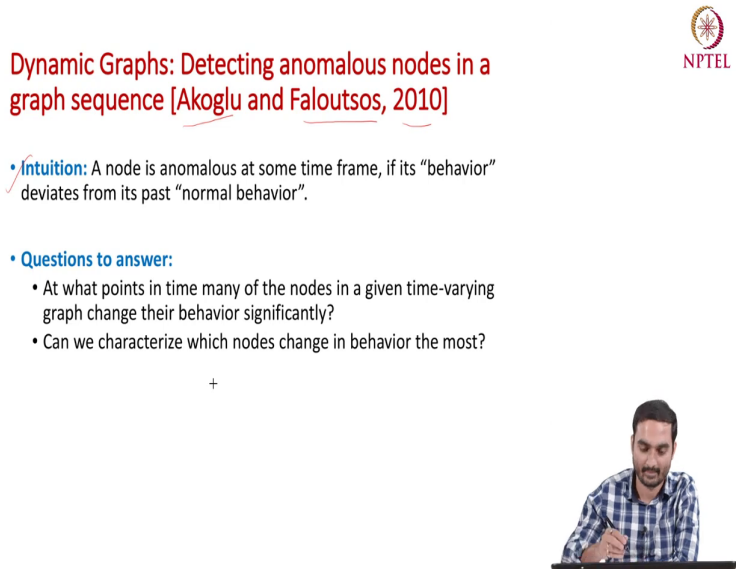
Hamming distance between the object between the agency matrix of the graph. So, counts of the number of graph entities in the matrix right, variations of edge weight distance you know lambda distance of adjacency matrix, this is basically 2 hop or you can take the Laplacian matrix.

And you know Laplacian matrix is basically defined by D minus A, where D is a degree, the diagonal this is basically diagonal matrix indicating the degree of nodes and this is a; this is adjacency matrix. So, you take the either the two hop or the Laplacian matrix and you

calculate the distance right. And then the diameter distance basically, the difference in the graph diameter right, defined in the greatest of the longest shortest path of all vertices.

So, at time stem t 0, you have a graph you take the diameter at timestamp t 1, you have the graph I mean the snapshot of the graph, you look at the diameter and then you measure the distance right. So, I mean you measure the difference between the diameters of two consecutive graph times, timestamp graphs right ok. So, these are the typical matrix that people generally use for measuring the distance between two consecutive functions ok.

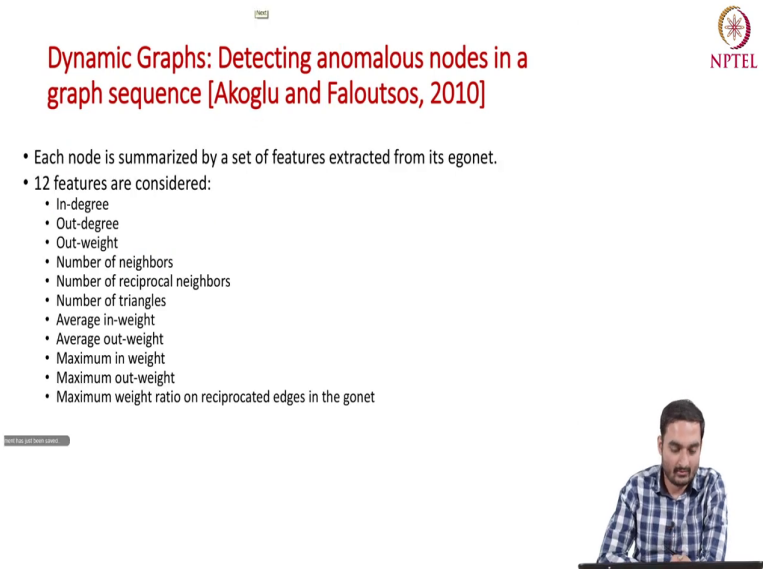
(Refer Slide Time: 12:04)



Let us look at one of the algorithms right which takes into account the dynamic graphs and detects anomalous entities. So, this is proposed by Leman Akoglu Christos Faloutsos in 2010. And what is the intuition? The intuition is that node is anomalous at some time frame, if its behavior deviates from its past normal behavior. So, we will quantify what is what do we mean by behavior, what do we mean by deviation and what do we mean by normal behavior ok.

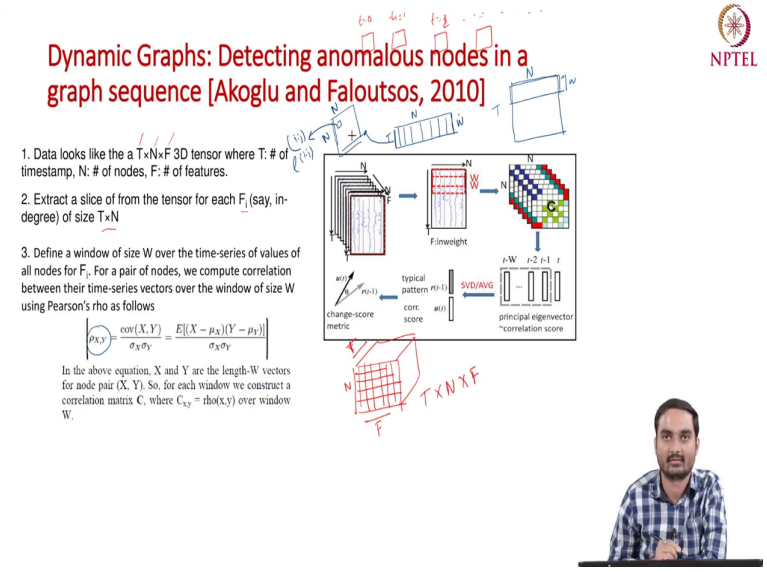
So, what are the questions? At what points in time many of the nodes in a given time varying graph change their behavior significantly? Can we characterize which nodes change in behavior the most ok?

(Refer Slide Time: 13:00)



So, for doing that, they extracted a set of features. Now, for every node right, they first identify the ego network one hop ego network and then they extracted these 12 features, in degree, out degree, out-weight number of neighbors and so on and so forth. Total 12 features, 12 features have been extracted for each node. So, extracted these features have been extracted from the ego network of that node, right.

(Refer Slide Time: 13:33)



Now, understand the problem statement. So, we have timestamp t equals to 0, t equals to 1, t equals to 3 2 and so on and so forth. We have graph timestamp, graph snapshots right and

every node has attributes ok. So, you can easily visualize it as a tensor, a 3D matrix where one dimension is time, other dimension is the number of nodes and other dimension is the features.

So, say this is number of nodes, every row indicates row node features and this is t right. So, this is a T cross N cross F tensor, ok. So now, the tensor you know tensor arithmetic is differ difficult. So, what they did? They extracted slice from the tensor, right. So, what they did? So, they fix a particular feature ok, let us say we have one feature F I, say the degree ok.

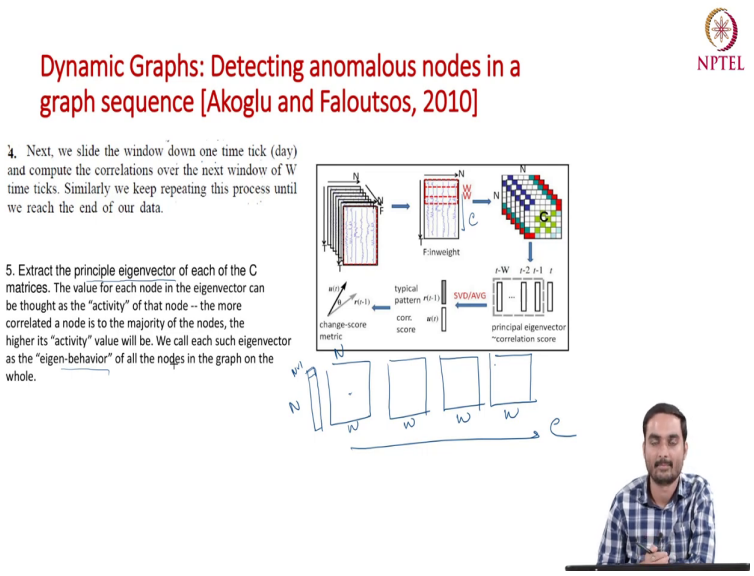
So, if we fix a particular dimension right, then we will get a 2D right. So, we get an N cross T matrix right, then what they did? They; so, you have this N cross T matrix right, then they defined a window of size W, right. So, what is the idea? So, this window within the window you basically measure the behavior of every node right and we define what do we mean by normal behavior, right.

Then we slide the window. So, this is the window, this is the next window, this is the next window and so on and so forth. You slide the window and at every window you measure the normal behavior of nodes ok. How do we do that? So, let us take, let us assume that you have a window W and this is and say let us say, ok. So, let me draw in a different manner. So, let us say this is N and this is T, this is W, ok.

So, you have a slice N, T and this time is essentially W right. And each column indicates nodes ok. Then for every node pair X, Y you measure the Pearson correlation coefficient ok. So, for every node pair X, Y you measure the Pearson correlation coefficient right. So, if within that window a pair of nodes behave similarly, then the Pearson correlation would be high otherwise low.

So, from every such sliding window, you get an N cross N matrix, because for every pair you are measuring the Pearson correlation. And each such entry indicates the correlation Pearson correlation between i and j right. Again you slide this W, you measure the Pearson correlation. So, for every W, for every slice you have this N cross N matrix, ok.

(Refer Slide Time: 18:07)



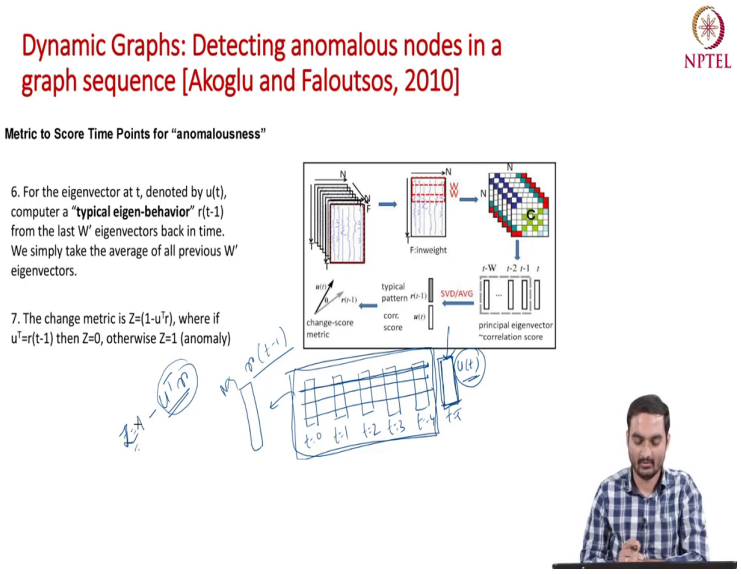
So, we slide the window down one time tick right and compute the correlations over the next window of W time ticks. You similarly keep on repeating the same process until you exhaust the data ok. So, you see here. So, let us assume that you have C, capital C such moving windows such slice ok. So, you have C such N cross N matrix; you have a C such N cross N matrices right.

Now, what you do? For every matrix you extract the principle eigenvector. So, principle eigenvector is kind of summarizes the activities at particular, at that time stamp, at that time window ok. So, for every so you have all these matrix matrices N cross N and you have, so window size W.

And how many such windows are there? C. So, for every such matrix you get a 1D vector which is the principle eigenvector, right. So, basically the idea is that so this would be N cross 1. So, every entry in that vector corresponds to the activity of that node, ok. So, I am reading it.

So, extract the principle eigenvector of each of the C matrices, the value for each node in the eigenvector can be thought as the activity of that node. The more correlated a node is to the majority of the nodes, the higher its activity value would be, because ultimately we have measured the similarity based on the Pearson correlation and then we extracted the principle eigenvector right. So, we call each such eigenvector, the eigen behavior of all the nodes in the graph as a whole, right.

(Refer Slide Time: 20:43)



So, this eigenvector kind of summarizes the activity at that particular time window, ok. So, let me draw it again. I am now, I am not drawing the N cross N matrix, but I am drawing the principle eigenvector right, right ok. So, now, let us focus on a particular time t; say t equals to 5 right. And let us say this is your u t, u t is the principle eigenvector of time t right.

So, we want to understand whether this summary is basically a summary how this summary is deviated from the previous summaries, right. So, I need to extract the summary of the previous timestamp timestamps right. So, what I can do? I can take all these principle eigenvectors, I can either do a element wise mean right or SVD right or PCA right and I get a squeezed representation of this another 1D, N cross 1.

If I just take a element wise mean, I get 1D vector and this is r of t minus 1, if this is t this is the summary till t minus 1 ok. Then what is the task? That the task is to see to measure the difference between this summary and this summary. How do we do that? Two vectors two 1D vectors, just take the dot product, right. So, u transpose r right. So, this is the similarity, 1 minus this is a difference right. So, and say this is z.

So, for every timestamp t we measure the z and higher the z value higher the anomalous behavior, because higher the z value lower the similarity right and higher the anomalous behavior. So, this is the idea. So, for every feature I get this z value, right. So, we have F such features right. So, we will have F such z values. Now, either we can aggregate or mean or whatever in some ways we aggregate and then identify the anomalous behavior, right.

So, this is one approach for detecting anomalous entities from an from a dynamic network right. Of course, you can think of your own ways, but this turned out to be really effective for you know small scale graphs, small scale dynamic graphs ok. So, I stop here. In the next lecture we will discuss the remaining you know remaining types of algorithms for anomaly detection. Particularly we look at dynamic graphs and you know some sort of community based approaches for anomalous entity detection ok.

Thank you.