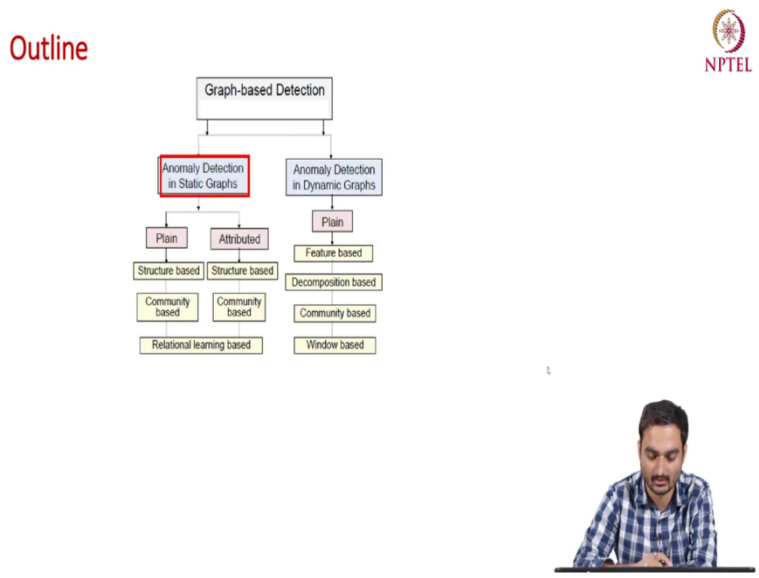
Social Network Analysis
Prof. Tanmoy Chakraborty
Department of Computer Science and Engineering
Indraprastha Institute of Information Technology, Delhi

Chapter - 08
Lecture - 02

So, in the last lecture we have tried to motivate you guys why you know anomaly detection is an important problem and what are the potential applications of anomaly detection in the context of Social Network and other context as well. So, in this lecture we will start off by you know by looking at a taxonomy of the algorithms that have been proposed for anomaly detection and we will and then we will look at some of the skeleton you know algorithms that that have already been there.

And then we will look at specifically few algorithms which have been considered as a state of the art as baselines not state of the art, but at least baselines in the context of anomaly detection ok.

(Refer Slide Time: 01:16)



Outline

Graph-based Detection

Anomaly Detection in Static Graphs

Plain

Structure based

Community based

Attributed

Structure based

Community based

Relational learning based

Anomaly Detection in Dynamic Graphs

Plain

Feature based

Decomposition based

Community based

Window based

So, this is the taxonomy as you see here in general we divide the algorithms into two categories static graphs. So, algorithms which are designed for detecting anomalous entities from a static graph and algorithms which are designed for detecting anomalies from dynamic graphs.

Now, if you look at the static graphs there are methods for plane static graph in which there is no attribute associated with nodes and edges and there are algorithms specifically designed for attributed static graphs right. And both the plane and attribute both for plane and attributed static graphs there are algorithms which only look at the structural properties of the network and there are algorithms which also look at the communities within a network right.

Basically, the idea is that you know we cannot detect outlier by looking at the global structure of a network rather we should focus on different sub parts of a network different clusters of a network and then with respect to a cluster we can define anomalous we can identify we can define anomalous entities and then identify them ok.

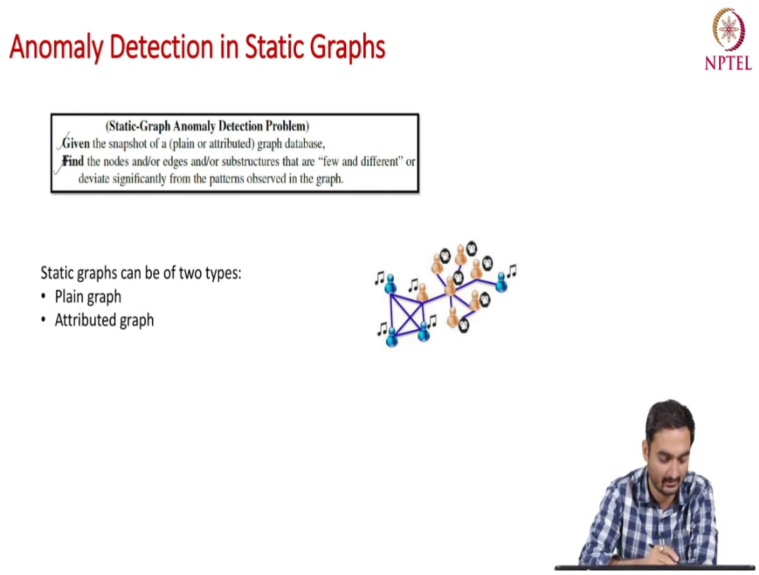
So, we will discuss one algorithm one such algorithm which will take care of the community structure and detect the communities as well as the anomalous nodes simultaneously right. And then we have algorithms which will which look which also look at the relational types for example, look at the relations between entities and based on that detect anomalous entities right.

With respect to the dynamic graphs we will look at again plane dynamic graphs feature based, decomposition based, how we can decompose a network into sub networks and look at the changes right of behaviour of networks into in subsequent time periods and based on that detect anomalous entities which should not have been there, but due to the some behavioural change you know those entities exist in the network. Here also we look at community based approaches and we look at window based approaches.

We in the dynamic network you know since we have static snapshots of a dynamic network at different points in time we fix a particular window and we see within the window what is the what is the normal behaviour and we then we keep moving the window and we will see that what is the abnormality across two subsequent windows and based on that we come up with some metrics and then we detect anomalous entities right.

So, I will not cover each of these types in this particular chapter rather I will give you an overview of each of these types and then I will focus on few specific types right. For example, I focus on one algorithm which was designed for community based approach community based outlet detection, I will focus on an algorithm which will look at dynamic graphs and based on window based approach detects outliers or abnormal entities right ok.

(Refer Slide Time: 04:44)



Anomaly Detection in Static Graphs

(Static-Graph Anomaly Detection Problem)
Given the snapshot of a (plain or attributed) graph database,
Find the nodes and/or edges and/or substructures that are "few and different" or deviate significantly from the patterns observed in the graph.

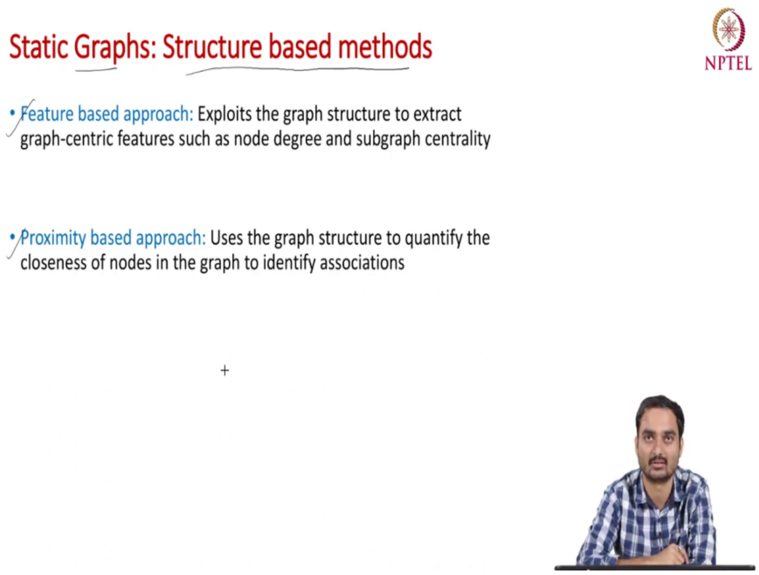
Static graphs can be of two types:

- Plain graph
- Attributed graph

So, let us look at the static graph ok. So, in the static graph what is the problem definition we are given a snapshot of a graph right a plain graph or an attributed graph and our task is to find nodes or and edges right or and substructures that are "few and different" or deviate significantly from the patterns observed in the graph ok and this is the again the broad definition, but we will try to make it concrete. And in this particular category we look at plane graphs and attributed graphs ok.

So, let us look at the plain graph and the algorithms which identify which extract structural based features from the plain graph and detect detects anomaly right.

(Refer Slide Time: 05:36)



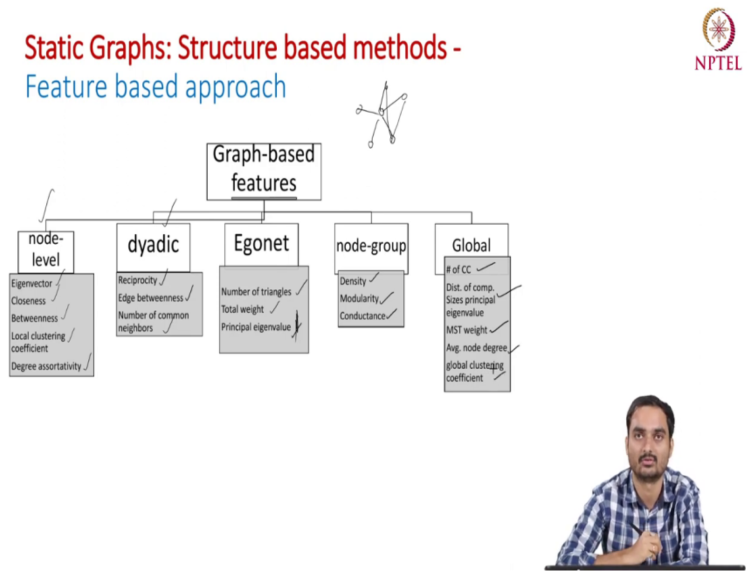
Static Graphs: Structure based methods

- Feature based approach: Exploits the graph structure to extract graph-centric features such as node degree and subgraph centrality
- Proximity based approach: Uses the graph structure to quantify the closeness of nodes in the graph to identify associations

So, if you look at the static graph and those algorithms which look at the structural based properties we will see two kinds of properties that are generally used. One is a simple feature based approach where we look at where we extract node centric features sub graph sub graph centric features and then we build normal classifiers.

And then in the other approach we define something called the proximity of nodes right in a graph and then depending on the sub proximities basically similarity between similarity or distance right between a pair of nodes and if a node is far from the rest of the nodes we can I mean we can you know categorize those nodes as anomalous nodes ok alright.

(Refer Slide Time: 06:33)



Static Graphs: Structure based methods -
Feature based approach

Graph-based features

node-level
- Eigenvector ✓
- Closeness ✓
- Betweenness ✓
- Local clustering coefficient ✓
- Degree assortativity ✓

dyadic
- Reciprocity ✓
- Edge betweenness ✓
- Number of common neighbors ✓

Egonet
- Number of triangles ✓
- Total weight ✓
- Principal eigenvalue ✓

node-group
- Density ✓
- Modularity ✓
- Conductance ✓

Global
- # of CC ✓
- Dist. of comp. Sizes principal eigenvalue ✓
- MST weight ✓
- Avg. node degree ✓
- global clustering coefficient ✓

So, let us start with a very simple approach feature based approach ok. So, in a normal graph based setting we look at five different types of features the first one is node level feature and we look at features like eigenvector centrality, closeness centrality, betweenness centrality, local clustering coefficient and degree associativity. Now, all these metrics have been discussed already right in the second chapter I guess ok. So, and all these features are node centric features.

So, what you basically do, you extract these features for individual nodes and then you can come up with some classifiers and these features you know would act as a normal attributes and then you and then you classify ok. If you move from node level to a to further coarse grained label we can look at dyadic structure you know properties of a pair of nodes right.

We can look at the reciprocity, reciprocity is again the extent to which you know two nodes are connected to each other say for example, if there is a node there is an edge from a to b dyadic edge whether there is another edge from b to a or not this is reciprocity.

We look at edge betweenness edge is basically a dyadic property right connects edge connects to one edge connects two nodes we look at edge betweenness centrality we also look at the number of common neighbours. We can look at say Jaccard coefficient between a pair of nodes we can look at you know the whole bunch of metrics that we discussed in the link prediction chapters Adamic - Adar distance and so on ok.

Then we move again from dyadic structure to you know some sort of Ego network structure where we look at ego network we already know what is a ego network. So, an ego network consists of a of an ego which is a node and it is one hop neighbours the induced subgraph of the one hop neighbours right. We look at the node and it is one hop neighbours and we also take their connections and that basically forms the egonet and then we extract different features from the egonet.

For example, we extract number of triangles, number of total weight right, principle eigenvector eigenvalue and so on. We will discuss these properties later on and that would determine the property of either a substructure or even the property of a node right of the ego for example, right.

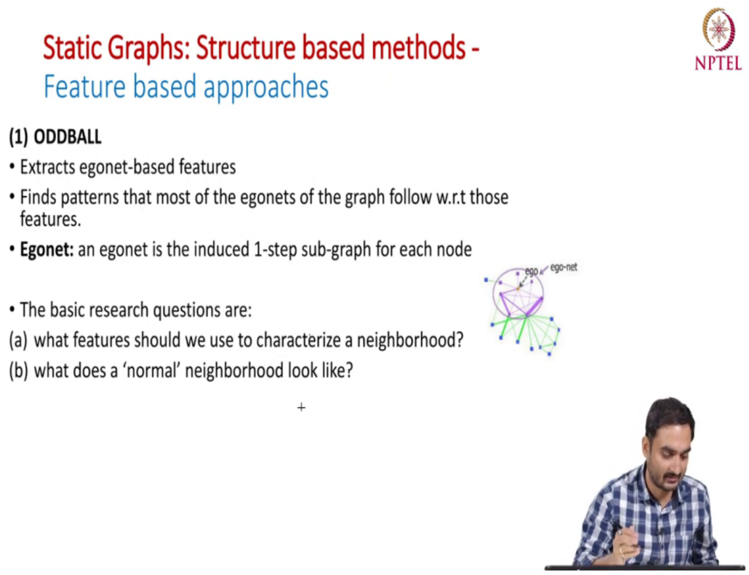
So, then we move from egonet to a community based or a group based property, where we look at say density of a community right actual number of connections divided by the possible number of connections. We look at Newman's modularity we can measure Newman's modularity, we can also measure conductance, we can measure permanence cut ratio and whole bunch of matrix that we discussed in the community detection chapter right.

And if we are also interested to look at the graph based properties right the overall graph based properties. We can measure a number of connected components right, we can measure the you know the distance of the distance the distance of different components and we can also look at the size of the components, we can look at the principal eigenvectors, we can look at principal eigenvalues, we can also look at the weights of a minimum spanning tree.

For example, the graph is weighted we can extract minimum spanning tree and from minimum spanning tree we can we check the weights of edges and we take the sum of weights of all the edges. We can also measure the, you know average degree of a node in the graph, we can also measure the global clustering coefficient right all these measures are global measures right.

And then we can simply employ classifiers like knife base or you know even logistic regression or support vector machine and classify nodes into anomalous and non anomalous and normal nodes right ok.

(Refer Slide Time: 11:01)



Static Graphs: Structure based methods - Feature based approaches

(1) ODDBALL

- Extracts egonet-based features
- Finds patterns that most of the egonets of the graph follow w.r.t those features.
- **Egonet:** an egonet is the induced 1-step sub-graph for each node

- The basic research questions are:

(a) what features should we use to characterize a neighborhood?

(b) what does a 'normal' neighborhood look like?

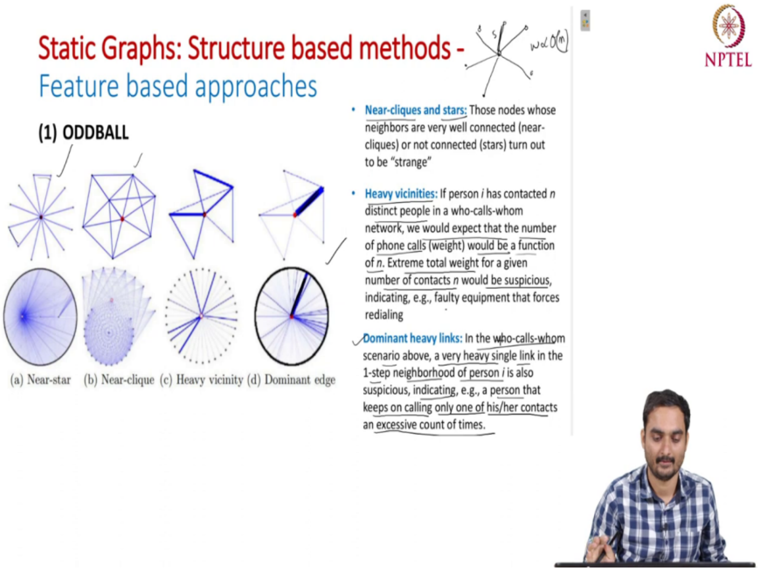
Now, let us look at the you know one such method which takes care of the features that we have discussed so far right, a subset of features that we have discussed so far and identifies anomalous entities ok. I think this is this is the, this is one of the first papers on anomaly detection in graphs right. This was published in PAKDD I think way back 2008 2009 by (Refer Time: 11:36) Christos Faloutsos and their group ok and this is one of the algorithms which detect anomalous entities in an unsupervised manner ok alright.

So, the name of this algorithm is called ODBALL right. What it basically does it? It basically extracts first it extracts egonet for each node you it takes a node and it is corresponding egonet and egonet constitutes the ego and the and only the one hop neighbours and the induced sub graph out of all these nodes. And then it basically extracts different measures of the egonet ok, we will discuss what kind of measures they generally use.

And then it basically tries to you know tries to look at the relations between these attributes of egonets ok. And they hypothesize that if a node is anomalous node right it is corresponding egonet structure exhibits a completely different properties right which is not there in a normal kind of ego network structure ok.

So, according to their paper they discover several new rules right we will discuss this rules right in the density, in the weights, in the ranks and the Eigen values that seem to govern right the so, called neighbourhood substructure which is the egonet right and they also show how to use this rules for anomaly detection right.

(Refer Slide Time: 13:27)



Static Graphs: Structure based methods -

Feature based approaches

$w \propto O(n)$

(1) ODDBALL

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

- Near-cliques and stars: Those nodes whose neighbors are very well connected (near-cliques) or not connected (stars) turn out to be "strange"
- Heavy vicinities: If person *i* has contacted *n* distinct people in a who-calls-whom network, we would expect that the number of phone calls (weight) would be a function of *n*. Extreme total weight for a given number of contacts *n* would be suspicious, indicating, e.g., faulty equipment that forces redialing
- Dominant heavy links: In the who-calls-whom scenario above, a very heavy single link in the 1-step neighborhood of person *i* is also suspicious, indicating, e.g., a person that keeps on calling only one of his/her contacts an excessive count of times.

So, now let us look at this algorithm carefully and this is very simple algorithm, but it turned out to be very useful in certain context ok. So, as I mentioned anomalous entities you know defining anomalous entities is challenging therefore, they simply started off by saying that look we are not interested to extract all possible anomalous entities rather we are interested to identify four types of anomalous entities right.

So, one is this cliques. So, clique or a near clique is basically anomalous because generally a graph does not constitute does not contain clique like structures. So, if we see a clique right it is basically an anomalous it may be considered as an anomalous entity. Similarly, star, the star like structure is again very rare right so, if we see a near star right it can again be considered as anomalous entity.

They also define something called heavy vicinity. So, let us look at this figure here right. So, you see that this is a near star right there is this central node and there are peripheral nodes right, but again this is with respect to the egonet ok. This egonet looks like a star, meaning that the peripheral nodes are not connected in that sense right.

You see there is only there are only two edges among the peripheral nodes right. Similarly, you see a near clique ok here this is the egonet node and one hop neighbours and almost all the neighbours are connected. The third n-th the third entity is called heavy vicinity ok, now what is heavy vicinity? So, they basically defined heavy vicinity with respect to an example right.

So, what they said I am quoting this thing from their paper, what they said is that if a person I mean let us think of a, who calls whom network ok. There are individuals and if somebody calls some other person there is an edge from that person to that person so, person x to person y ok who calls whom network.

So, in the who calls whom network if person I has contacted n distinct person n distinct people in the who calls whom network we would expect that the number of phone calls which is essentially the weights right, think about it. So, say this is the ego and these are the altars right peripheral nodes and this guy is calling these guys ok and if and the weight associated with an edge indicates the number of times ego has called an altar right say this is 5 means this guy has called this guy 5 times and so on and so forth.

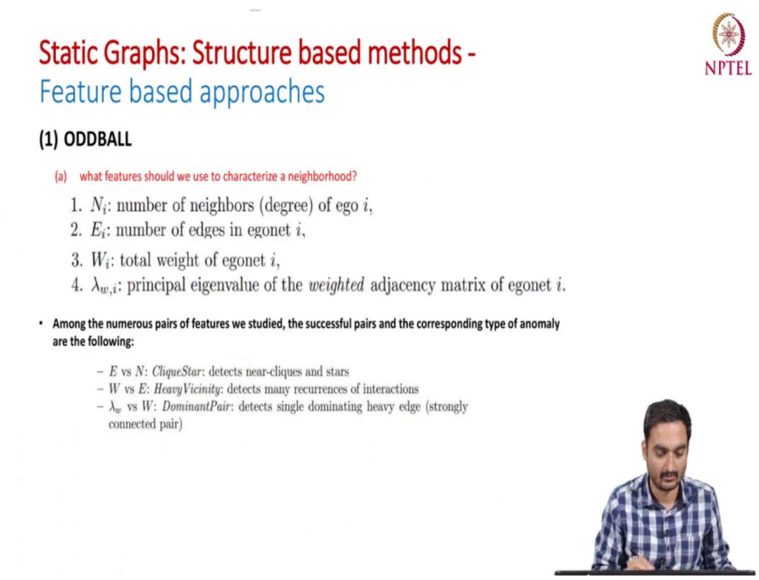
So, right we would expect that the number of phone calls which is the weight right would be a function of n right. So, the weight is it should be order of order of n, n is the number of number of peripheral nodes number of alters ok or you can think of it as number of nodes in the egonet because basically n plus 1 is order of n right.

So, then they said that extreme total weight for a number of contacts n would be suspicious right, think about it. Say you see that this guy has called lots of times to a subset of his alter nodes his neighbours right lots of times which is not a simply an order of n right that is suspicious right. For example in who calls whom it can be just a faulty equipment which basically forces redialling right and therefore, called you know calls have happened automatically right. So, this is called heavy vicinities.

So, the fourth category is called dominant heavy links, this is again related to who calls whom network now we are looking at individual weights of edges and if you see that one edge has a lot of weight right compared to the other edges then there is some problem ok. So, in a who calls whom scenario, a very heavy single link right in the one step neighbourhood of a person i is also suspicious ok, indicating for example, a person that keeps on calling only one of his or her contacts an excessive count of times ok.

So, this is the dominant heavy link you see that this link is very heavy ok and therefore, the edge weight would be a lot edge weight would be much higher than the remaining edge weights ok. So, they wanted to detect these four types of categories ok alright.

(Refer Slide Time: 19:13)



Static Graphs: Structure based methods - Feature based approaches

(1) ODDBALL

(a) what features should we use to characterize a neighborhood?

1. N_i: number of neighbors (degree) of ego i,
2. E_i: number of edges in egonet i,
3. W_i: total weight of egonet i,
4. $\lambda_{w,i}$: principal eigenvalue of the *weighted* adjacency matrix of egonet i.

- **Among the numerous pairs of features we studied, the successful pairs and the corresponding type of anomaly are the following:**
  - E vs N: *CliqueStar*: detects near-cliques and stars
  - W vs E: *HeavyVicinity*: detects many recurrences of interactions
  - λ_w vs W: *DominantPair*: detects single dominating heavy edge (strongly connected pair)

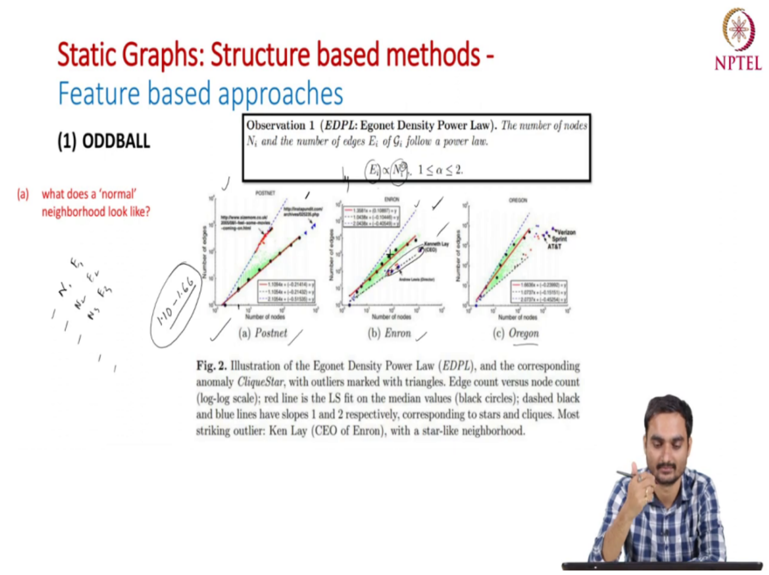
So, now let us look at what are the features that they have considered? They have considered remember they have only looked at the ego network right and they have considered the number of neighbours ok which is basically the degree of the ego. Then the number of edges in the ego network it is not that ego it is not the degree of the ego because it is the induced sub graph neighbours are also connected.

So, E i is the number of edges and then we have you know total weight of the egonet which is sum of the weights of all the edges. And the fourth one lambda is the principal eigenvalue of the weighted adjacency matrix of egonet i right. You have this small egonet you can get the adjacency matrix weighted adjacency matrix you can get the principle eigenvector and the corresponding principal eigenvalue right.

So, these only these are the properties that they have considered and what they claimed is very interesting. They claimed that if you look at the relationship between E and N right, E and N that would give you that would help you identify cliques and stars ok. Secondly, if we look at E and W right that would help you identify heavy vicinity kind of anomalous entities ok.

If we look at W and lambda right that would help you identify dominant pairs right. You may wonder how have I mean how they have you know concluded these things right let us discuss ok.

(Refer Slide Time: 21:09)



Static Graphs: Structure based methods - Feature based approaches

(1) ODDBALL

(a) what does a 'normal' neighborhood look like?

Observation 1 (*EDPL*: Egonet Density Power Law). *The number of nodes N_i and the number of edges E_i of $\mathcal{G}_i$ follow a power law.*

$$E_i \propto N_i^{\alpha}, \quad 1 \leq \alpha \leq 2.$$

(a) *Postnet* (b) *Enron* (c) *Oregon*

Fig. 2. Illustration of the Egonet Density Power Law (*EDPL*), and the corresponding anomaly *CliqueStar*, with outliers marked with triangles. Edge count versus node count (log-log scale); red line is the LS fit on the median values (black circles); dashed black and blue lines have slopes 1 and 2 respectively, corresponding to stars and cliques. Most striking outlier: Ken Lay (CEO of Enron), with a star-like neighborhood.

So, the first observation is you know is that if you look at the right if you look at the relationship between number of nodes and the number of edges in a in an egonet it should follow a Power Law right. So, according to them E i should be proportional to N i to the power alpha where in a normal case alpha should vary between 1 and 2 ok.

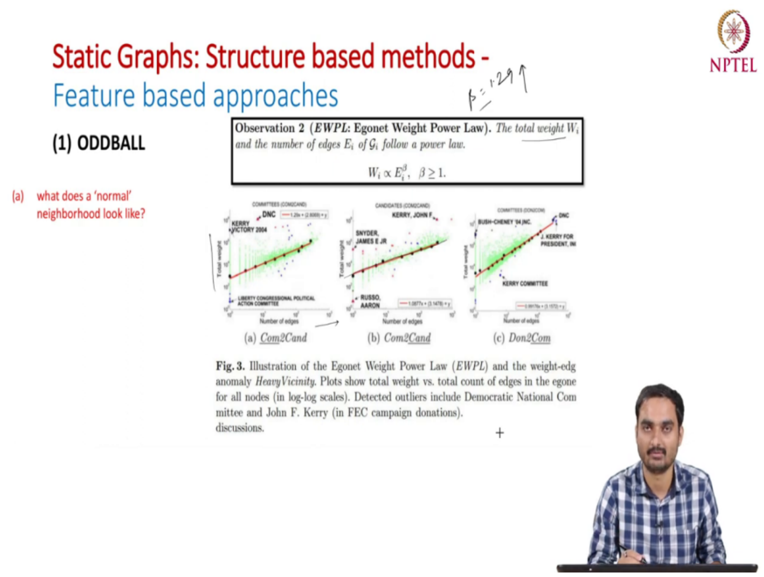
So, right and in a normal case what they said, they said that although alpha varies between 1 and 2, but the ideal range is 1.10 to 1.66 right. If you see any entity which does not belong to this range they are suspicious ok. So, let us look at an example this one right.

So, they have tested this on multiple data sets the one is called Postnet data set, the other is this Enron data set and the third one is Oregon data set. I am not explaining the, you know data set properties and all you can look at the paper, but I am just you know telling you the observation the major observation ok alright.

So, what is the first observation? For example, one entity was right one entity was Ken Lay who was the CEO of Enron right; obviously, he is an exception. So, he is an outlier right. So, there is no ground truth as such, but they tried to evaluate their method based on some manually inspecting some of the entities which do not belong to the standard deviation or the normal range that should be there right in the in this particular case.

So, what they said is that if you look at if you look at entities which are here right which fall even below the slope 1 or just above the slope 1 they are kind of star like structure right and if some if someone lies some point lies on the slope 2 right or above much above the standard deviation the that entity may look like you know the clique like entity ok. So, in this way you can identify cliques and stars alright.

(Refer Slide Time: 24:05)



Static Graphs: Structure based methods -
Feature based approaches

(1) ODDBALL

(a) what does a 'normal' neighborhood look like?

Observation 2 (*EWPL*: **Egonet Weight Power Law**). *The total weight W_i and the number of edges E_i of $\mathcal{G}_i$ follow a power law.*

$$W_i \propto E_i^{\beta}, \quad \beta \geq 1.$$

(a) Com2Cand (b) Com2Cand (c) Don2Com

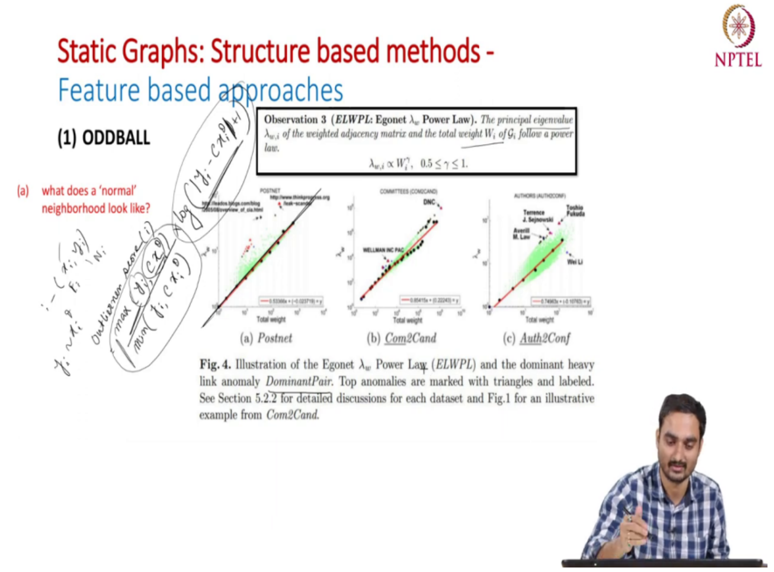
Fig. 3. Illustration of the Egonet Weight Power Law (*EWPL*) and the weight-edg anomaly *HeavyVicinity*. Plots show total weight vs. total count of edges in the egone for all nodes (in log-log scales). Detected outliers include Democratic National Com mittee and John F. Kerry (in FEC campaign donations).
discussions.

Let us look at the second observation; in the second observation they try to differentiate a heavy vicinity from the remaining population. And what they hypothesized? They hypothesized that total weight right and the number of edges E i right these two features should follow a power law ok. And according to them you know this power law this beta is you know this beta should basically be beta should be 1.29 right.

So, the range beta should be up to 1.29 if it is above 1.29 then there is some problem ok and in the similar manner they plotted the number of edges in the x axis and number of weights on the y axis right, you see this black dots black circles the best fit the best fitted line red line standard deviations and outlier points right.

Again their conclusion was kind of same that those points which are far from this standard deviation they are they are most probably outliers ok I mean outliers and their type is essentially heavy vicinity ok.

(Refer Slide Time: 25:29)



Static Graphs: Structure based methods -

Feature based approaches

(1) ODDBALL

(a) what does a 'normal' neighborhood look like?

Observation 3 (*ELWPL*: **Egonet λ_w Power Law).** *The principal eigenvalue $\lambda_{w,i}$ of the weighted adjacency matrix and the total weight W_i of $\mathcal{G}_i$ follow a power law.*

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

$i - (x_i, y_i)$ E_i N_i

$y_i \propto x_i^{\theta}$

Outlierness score (i)

$$\frac{\max(y_i, Cx_i^{\theta})}{\min(y_i, Cx_i^{\theta})} \times \log(|y_i - Cx_i^{\theta}| + 1)$$

(a) *Postnet* (b) *Com2Cand* (c) *Auth2Conf*

Fig. 4. Illustration of the Egonet λ_w Power Law (*ELWPL*) and the dominant heavy link anomaly *DominantPair*. Top anomalies are marked with triangles and labeled. See Section 5.2.2 for detailed discussions for each dataset and Fig.1 for an illustrative example from *Com2Cand*.

So, let us look at the third observation, what they said is that there is again a kind of power law relation between the principal eigenvalue lambda of the egonet of the weighted adjacency matrix of the egonet and the total weight W i ok. And basically the again the idea is same you plot it and if you see that the points are you know far from the standard deviation they possibly indicate that the structure is a kind of a heavy dominant pair like structure ok.

So, visually this is very you know very very interesting right to look at, but how do we do this thing automatically ok. So, what they proposed is that you have all these features right. So, you have four basically features and you have all combinations of features right x i x i y i ok. So, what they defined?

They defined something called outlier outlierness score right. So, outlierness score of a particular node or the or a particular egonet does not matter same right is defined by max y i comma C x i theta by mean y i comma C x i theta times log mod y i minus C x i theta mod plus 1.

Let me explain what it is right. So, for every node i you have you have a pair of features there are four features and you can have 4 C 2 pairs right. Let us say one such pair is x i and y i one such pair say x i is E i number of edges and y i is N i number of nodes ok, then what you do?

So, and you know normally y i and x i should follow power law right. So, y i should be proportional to x i to the power theta, theta is the coefficient depending on you know this observation that we mentioned theta can be anything theta can be theta can have certain range right.

So, C times x i C times x i theta x i to the power theta right this is what, this is basically the fitted value right, since you best fit using a line right log scale, remember in the log scale power law behaves a line right power lies basically behaves like a line. So, in a log scale C x i is the projected value right the best fit value right and what is y i, y i is the actual value right. So, essentially y i minus C x i theta is the error right.

Now, in this particular formula you see that there are two times that, we are penalizing, if there is an error this is one component of the error and this is another component of the error right. This is a simple max min normalization right, the higher the deviation the higher the numerator right and this is again another error component, but in a log scale right and the higher the value the higher the outlier outlierness score is not it.

This is one way of capturing the deviation right you can have another way of capturing the deviation, but this is a very simple way very effective scalable right unsupervised therefore, they proposed this approach ok. And then you do not need to look at it visually, you just calculate the calculate this value for each entity and then you rank all these entities based on this outlierness score and then you return entities whose outlierness scores are higher ok.

So, this is about ODDBALL, I strongly suggest you guys to read the paper fantastic paper, but you know lots of interesting ideas, but those ideas may not be useful in today’s context I mean in the context of deep learning and stuff. But this was really a nice paper; I think this paper got the Test of Time Award in the last year PAKDD if I am not mistaken ok, this is really a fantastic paper.

So, you can also look at. So, that was basically a feature based approach right in the next lecture we will look at proximity based approach right where we see you know how these two nodes are related how they are different distance wise and so on and so forth and based on that we will try to identify outliers ok. With this I stop this lecture today and then in the next lecture we will discuss the following part of the chapter.

Thank you.