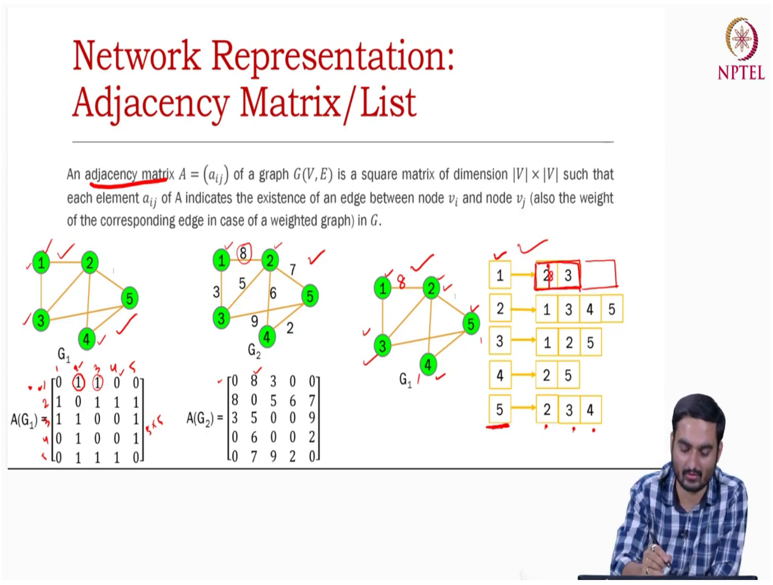
Social Network Analysis
Prof. Tanmoy Chakraborty
Department of Computer Science and Engineering
Indraprastha Institute of Information Technology, Delhi

Chapter - 01
Lecture - 02

Now, we will discuss how we can store a network, right. Now, so far we have discussed about different types of networks, but computer does not know right you know all these things. So, how do we you know let the computer know that look this is the structure of a network? And the easiest way to represent a network or a graph you all know is basically through adjacency matrix ok.

Now, what is adjacency matrix? Adjacency matrix is a matrix where rows correspond to different nodes also columns correspond to the same set of nodes, right. And if and an entry in the matrix, if the entry is 1 meaning that these two nodes are connected, right.

(Refer Slide Time: 01:07)



Network Representation: Adjacency Matrix/List

An adjacency matrix $A = (a_{ij})$ of a graph $G(V,E)$ is a square matrix of dimension $|V| \times |V|$ such that each element a_{ij} of A indicates the existence of an edge between node v_i and node v_j (also the weight of the corresponding edge in case of a weighted graph) in G.

$$A(G_1) = \begin{bmatrix} 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 1 & 0 \end{bmatrix}$$

$$A(G_2) = \begin{bmatrix} 0 & 8 & 3 & 0 & 0 \\ 8 & 0 & 5 & 6 & 7 \\ 3 & 5 & 0 & 0 & 9 \\ 0 & 6 & 0 & 0 & 2 \\ 0 & 7 & 9 & 2 & 0 \end{bmatrix}$$

$1 \rightarrow 2, 3$
$2 \rightarrow 1, 3, 4, 5$
$3 \rightarrow 1, 2, 5$
$4 \rightarrow 2, 5$
$5 \rightarrow 2, 3, 4$

Let see an example here. So, this is a graph G1 or a network G1. And let us say this is right this row correspond to node 1, this is 2, 3,4, 5. Similarly, 1, 2, 3, 4, 5. So, you have a 5 cross 5 matrix. And you see that this entry is 1 ok. Meaning, node 1 and node 2 they are connected ok. Node 1 and node 3 are also connected therefore, this is 1, right. Node 1, node 4 are not connected therefore, this is 0, right. So, in this way we can create an adjacency matrix, but this is a small network right only a 5 nodes.

So, think of a node with billions of nodes. So, you can have billion cross billion matrix, right. This is very very difficult to store. It is extremely space consuming more importantly if you store such a large network using adjacency matrix. The matrix would be sparse extremely sparse, because most of the nodes are not connected, right; therefore, you will see a lot of 0 entries in the matrix which is not at all useful, right.

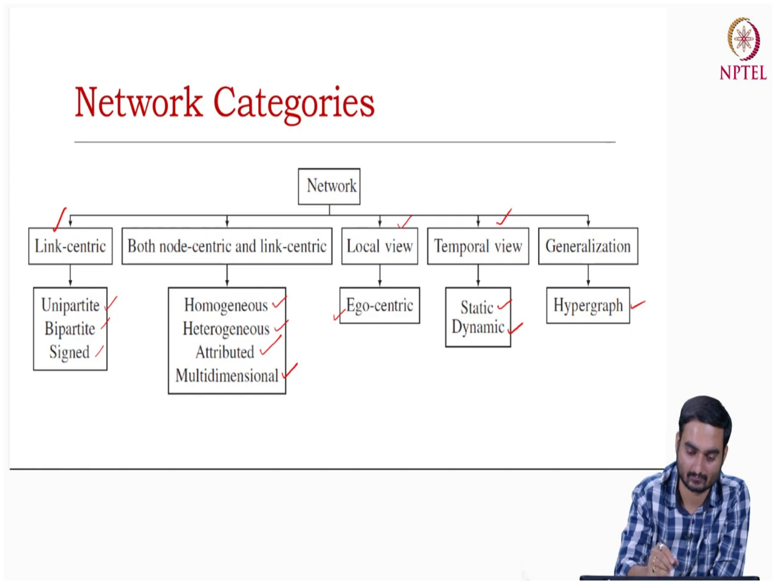
So, how can we come up with even efficient way of representing network? Definitely, you know another way is through something called adjacency list, right. In a data structure there is a there is a data structure called linked list, right. And it basically says that right that you know. So, for every node you have a link list ok. So, for example, node 1 right, you see 1 is connected to 2 and 3.

Therefore a link list is formed with node 2 and 3. Therefore, you see, right. So, and so for say 4 and 5, you do not need to create any other cells, right. So, these two spaces are basically be useful now can be useful. Similarly, for say node 5, node 5 is connected to 2, 3 and 4. You see here corresponding to node 5 you have 2, 3 and 4, right. In this way you basically can ignore 0 entries and you can get a compact representation.

Now, this is for unweighted network; if you have a weighted network like this edges are associated and weights are associated with edges in that case your entries in the adjacency matrix will be the weights, right. For example, you see here this is also a 5 cross 5 matrix and node 1 and node 2 they are connected through this edge whose weight is 8, you see that this is 8, right.

Similarly, you can do the same thing in case of adjacency you know list where with every associated neighbor because 2 is a neighbor of 1, you can also create another cell right in the link list which indicates the weight, right. For example, say if 1 and 2 are connected through an edge of weight 8, you can write 8 here, right. So, this is very well known and therefore, I hope you understand it, right.

(Refer Slide Time: 04:27)



Network Categories

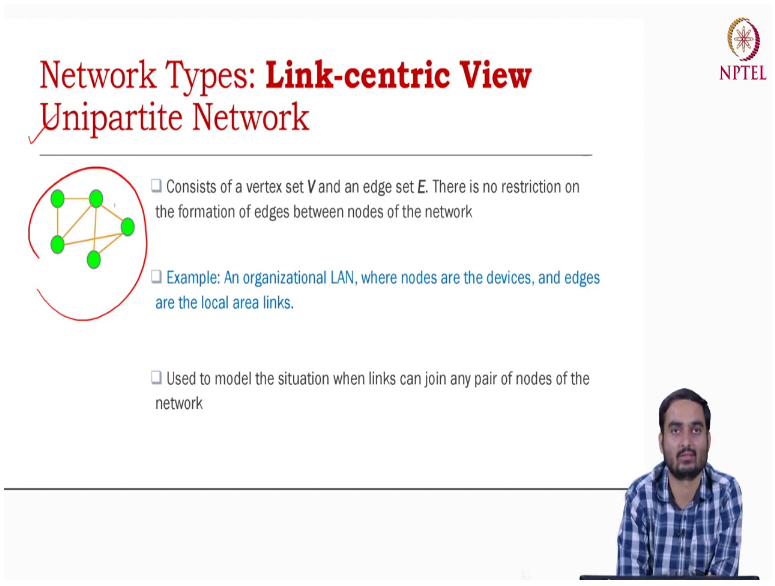
- Network
  - Link-centric
    - Unipartite
    - Bipartite
    - Signed
  - Both node-centric and link-centric
    - Homogeneous
    - Heterogeneous
    - Attributed
    - Multidimensional
  - Local view
    - Ego-centric
  - Temporal view
    - Static
    - Dynamic
  - Generalization
    - Hypergraph

So, let us see how you can categorize different types of networks. Well, there are different ways to categorize networks, right; this is one way to categorize it. So, here we categorize a network based on the node and link structure, right. So, only based on link centric view, we will see three kinds of networks, one is called Unipartite network, the second one is Bipartite and the Signed network.

Based on both nodes and links you can think of four types of networks Homogeneous network, Heterogeneous network, Attributed network and Multi dimensional network. Based on a local view of a network you can think of something called Egocentric network which is very locally focused, right. You can also think of a temporal network if the time dimension is available in the network, right.

It can be if the time dimension is not available it can be static network, if the time dimension is available it can be a dynamic network. And of course, you can generalize the network structure in a more generic way it is basically called Hypergraph. We will discuss each and every category separately, right.

(Refer Slide Time: 05:39)


Network Types: **Link-centric View**

Unipartite Network

❑ Consists of a vertex set *V* and an edge set *E*. There is no restriction on the formation of edges between nodes of the network

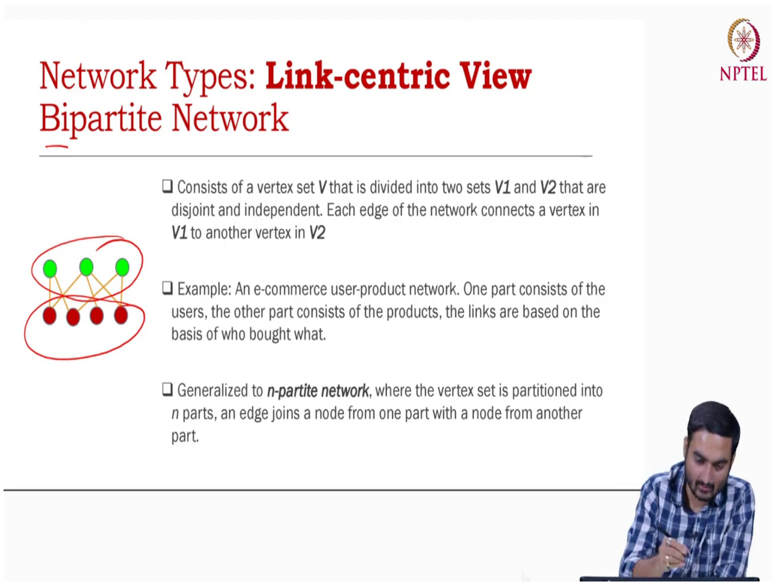
❑ Example: An organizational LAN, where nodes are the devices, and edges are the local area links.

❑ Used to model the situation when links can join any pair of nodes of the network

So, let us look at link centric view ok. So, and the first category is unipartite network. In a unipartite network there is only one partition. Now, what do you mean by partition? It will be it will be clear in the next slide, right. So, in this particular example you see that there is no partition as such meaning that, nodes are not separated, right.

Nodes are connected right and there is no particular constraint for example, there is no there is no constraint that ok these two nodes should not be connected or these two nodes should be placed in different partitions, right. This is a unipartite network which is a general kind of network that we generally talk about.

(Refer Slide Time: 06:25)



Network Types: **Link-centric View**
Bipartite Network

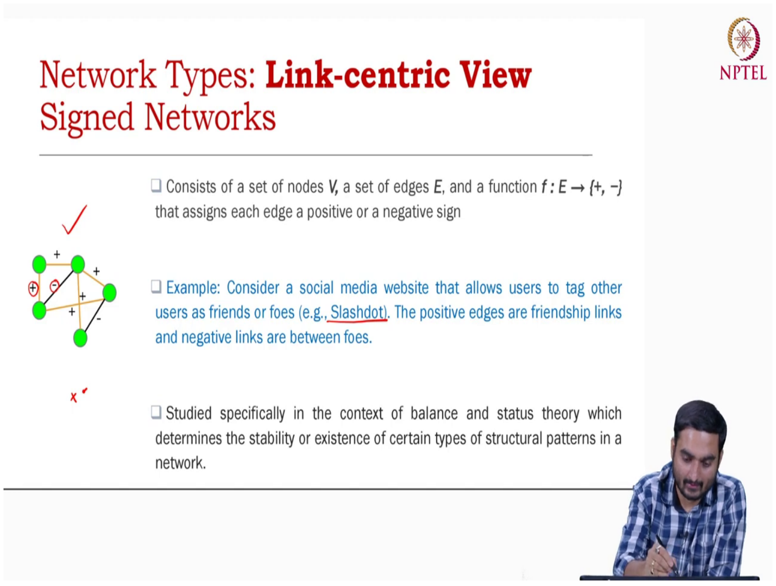
- Consists of a vertex set *V* that is divided into two sets *V1* and *V2* that are disjoint and independent. Each edge of the network connects a vertex in *V1* to another vertex in *V2*
- Example: An e-commerce user-product network. One part consists of the users, the other part consists of the products, the links are based on the basis of who bought what.
- Generalized to ***n-partite network***, where the vertex set is partitioned into *n* parts, an edge joins a node from one part with a node from another part.

The next one is called bipartite network ok. As the name suggests it has two partitions, right. You see here this is one partition this is another partition ok. The partition with color green right you see that within a partition nodes are not connected, right. So, green nodes are not connected red nodes are also not connected.

But across two partitions nodes are connected, right. Now, this is a typical nature of bipartite network you can think of right you can think of say E-commerce network for example, where you have users in one partition you have items in another partitions right if user if one user has bought a particular item you can connect that user to that item, right.

But users are not connected because users cannot buy each other. At least in our country right users cannot buy each other. Similarly, in the other partition item partition items can also cannot buy each other. So, items are not connected ok. So, this is called bipartite network. You can think of tripartite network for example, multipartite network for example, the only constraint is that within a partition nodes should not be connected, nodes can only be connected across partitions ok.

(Refer Slide Time: 07:47)



Network Types: **Link-centric View**
Signed Networks

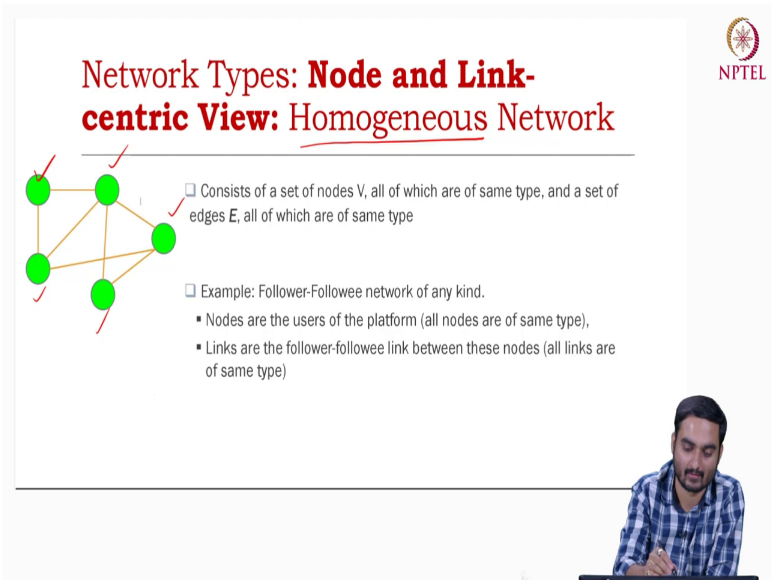
- Consists of a set of nodes V, a set of edges E, and a function $f: E \rightarrow \{+, -\}$ that assigns each edge a positive or a negative sign
- Example: Consider a social media website that allows users to tag other users as friends or foes (e.g., Slashdot). The positive edges are friendship links and negative links are between foes.
- Studied specifically in the context of balance and status theory which determines the stability or existence of certain types of structural patterns in a network.

So, the third link centric view is signed network ok. You see here in this particular network, nodes are connected it can be unipartite, it can be bipartite, but edges are associated with signs, right. The signs can be positive negative and so on. The signs indicate the types of relations between users. For example, if two users are friend versus two users are enemy, right.

So, if two users are friend you can add a positive sign, if two users are enemy then you can add a negative sign. You can think of you know social network like Slashdot or epinion, right. These social networks you can explicitly mention that these are the users who I have I have friendship these I these are the users who I have animosity for example, right they are your enemies, right.

So, depending on that you can assign signs. In fact, in the citation network for example, right the one that I mentioned in the last lecture if a paper criticizes another paper, if a paper criticizes another paper you can actually think of a negative link, a negative link between two papers if a paper you know appreciates another paper, you can think of a positive link, right.

(Refer Slide Time: 09:15)

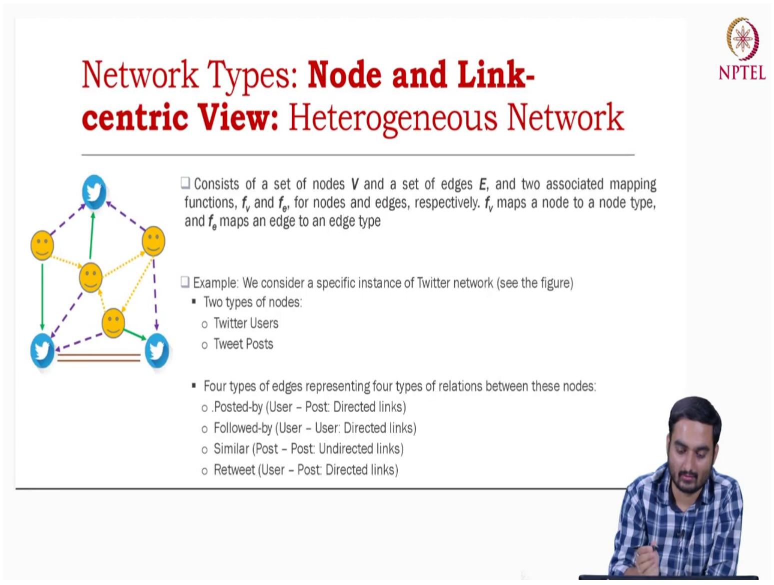


Network Types: **Node and Link-centric View:** Homogeneous Network

- Consists of a set of nodes V, all of which are of same type, and a set of edges *E*, all of which are of same type
- Example: Follower-Followee network of any kind.
  - Nodes are the users of the platform (all nodes are of same type),
  - Links are the follower-followee link between these nodes (all links are of same type)

So, now let us look at both node and link centric view ok. If you look at both node and link together, you can think of a network called homogeneous network, right. In the homogeneous networks, nodes are of same types edges are of same types, right. For example, user-user interaction network say twitter follower-followee network where users are you know nodes are users right and links are follower-followees. It is not the case that one node is user, another node is tweet no, right. Here, all users are of same types links can be also of same types ok.

(Refer Slide Time: 09:55)



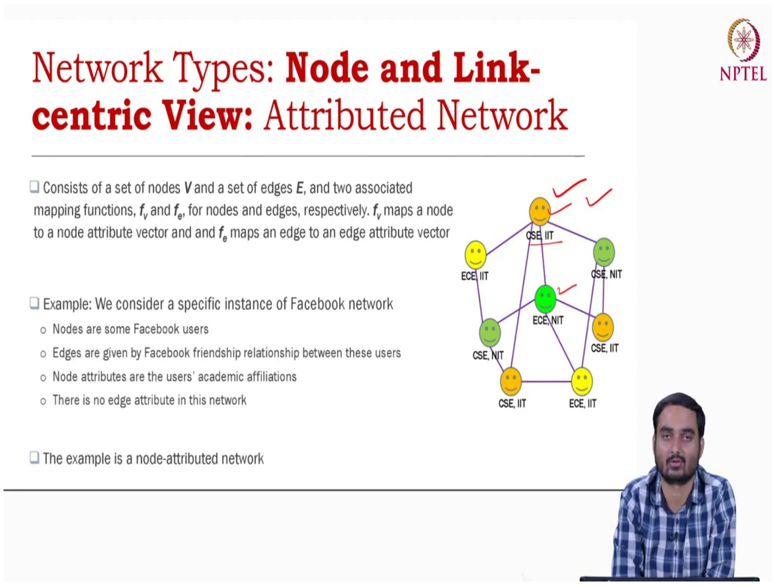
Network Types: **Node and Link-centric View:** Heterogeneous Network

- Consists of a set of nodes *V* and a set of edges *E*, and two associated mapping functions, f_v and f_e, for nodes and edges, respectively. f_v maps a node to a node type, and f_e maps an edge to an edge type
- Example: We consider a specific instance of Twitter network (see the figure)
  - Two types of nodes:
    - Twitter Users
    - Tweet Posts
  - Four types of edges representing four types of relations between these nodes:
    - Posted-by (User – Post: Directed links)
    - Followed-by (User – User: Directed links)
    - Similar (Post – Post: Undirected links)
    - Retweet (User – Post: Directed links)

On the other hand, if you think of heterogeneous networks you can imagine different types of users, right. Say, one type of users correspond to one type of nodes corresponds to you can think of different types of nodes. One type of nodes can be can be users, another type of nodes can be tweets, right. There can be a link from a user to a tweet through the link called posted by.

There is a link between a user to another user through follower-followee and so on and so forth. So, depending upon the heterogeneity with respect to the nodes or with respect to the edges you can think of node heterogeneous graph, edges, edge heterogeneous graph or both node and edge heterogeneous graph, right.

(Refer Slide Time: 10:46)



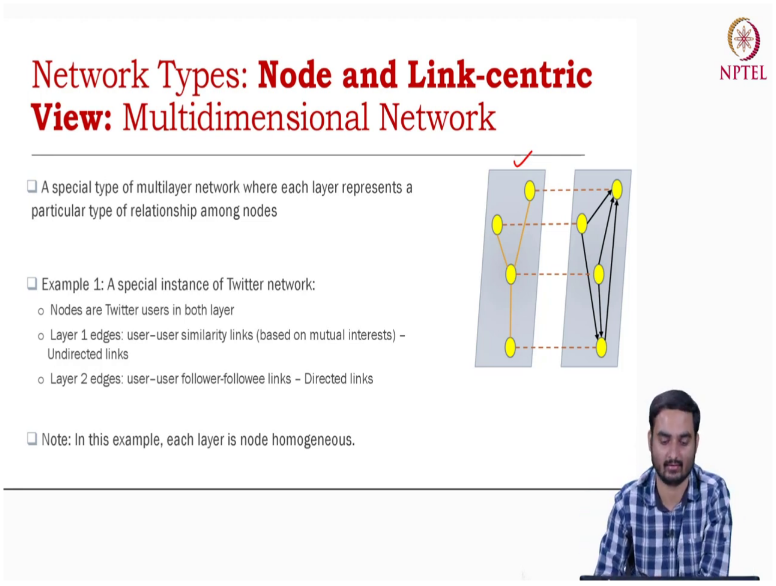
Network Types: **Node and Link-centric View:** Attributed Network

- Consists of a set of nodes *V* and a set of edges *E*, and two associated mapping functions, f_v and f_e, for nodes and edges, respectively. f_v maps a node to a node attribute vector and and f_e maps an edge to an edge attribute vector
- Example: We consider a specific instance of Facebook network
  - Nodes are some Facebook users
  - Edges are given by Facebook friendship relationship between these users
  - Node attributes are the users' academic affiliations
  - There is no edge attribute in this network
- The example is a node-attributed network

And the third type is called attributed network, where you know where you can see here in this particular example where you know nodes can be associated with attributes; edges can also be associated with attributes. For example, you see now this is the interaction networks of students right where this student belongs to CSE, right some IIT these student belongs to ECE department some NIT and so on and so forth, right.

These are different features meta information about users these are attributes, right. And these attributes are very important because you know in one of the chapters later we will discuss something called link prediction, right, where we will see how this based on these attributes we can actually predict that which edge is going to form in the near future, right. This is called link prediction or this is link this is also called link recommendation.

(Refer Slide Time: 11:44)



Network Types: **Node and Link-centric View:** Multidimensional Network

- A special type of multilayer network where each layer represents a particular type of relationship among nodes
- Example 1: A special instance of Twitter network:
  - Nodes are Twitter users in both layer
  - Layer 1 edges: user–user similarity links (based on mutual interests) – Undirected links
  - Layer 2 edges: user–user follower-followee links – Directed links
- Note: In this example, each layer is node homogeneous.

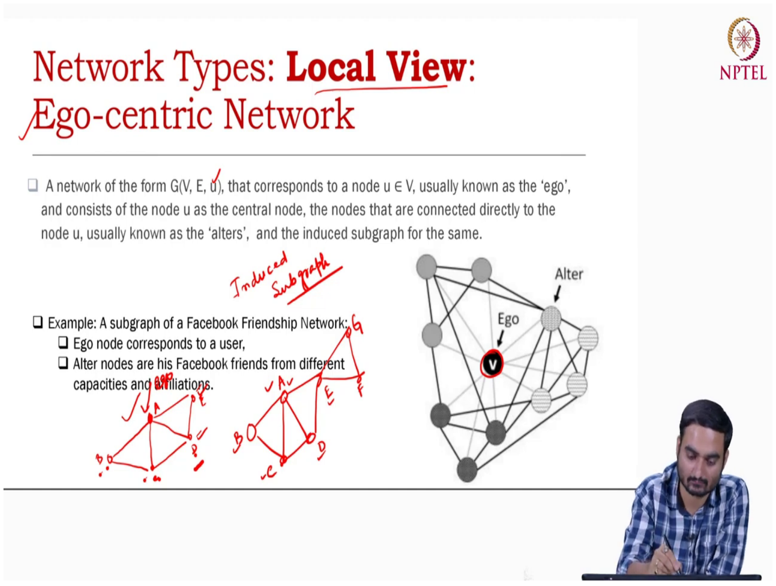
So, there is another type of you know node a centric view which is called Multi dimensional network. This is again a little bit generic kind of structure where you can think of different you know different layers in a network; for example, say again let us take twitter right you can have nodes indicating different users in one layer nodes are connected through follower-followee links.

In another layer nodes are connected through say re tweet link, in another layer nodes are connected through re share link and so on and so forth. So, you can you may argue that you know why we need a multi dimensional network, you can also capture such heterogeneity based on heterogeneous network yes you can do that, but sometimes what happens is that we desperately need that a network becomes homogeneous, right.

Because, when a network becomes homogeneous whole bunch of tasks becomes so easier, right. So, the idea is that at every layer you can think of it as a homogeneous network. So, you have homogeneous network of users, users are connected to follower-followee only one type of link. The second layer you have users links are re tweets, right.

And the third is another you know homogeneous network where nodes are users and links can be you know sharing or coating and so on and so forth, right. So, every layer is homogeneous, but if you look at the compact network the entire network, this is basically heterogeneous.

(Refer Slide Time: 13:29)



Network Types: **Local View**: Ego-centric Network

- A network of the form G(V, E, u), that corresponds to a node u ∈ V, usually known as the 'ego', and consists of the node u as the central node, the nodes that are connected directly to the node u, usually known as the 'alters', and the induced subgraph for the same.
- Example: A subgraph of a Facebook Friendship Network:
  - Ego node corresponds to a user,
  - Alter nodes are his Facebook friends from different capacities and affiliations.

So, now, let us look at local view ok. So, now, we basically you know try to look at a particular portion of a network, put a lens and magnify it and try to understand what is there. So, in a local view of a network you can think of something called egocentric network.

What is ego centric network? So, in a egocentric network is a node centric view for every node you can create a egocentric network ok. How? So, let us say let us say this is a network ok and say this is A right, this is B, C, D, E, F and G. These are nodes, right. And I want to create an ego network for node A, right. So, ego network consists of two types of nodes; one is the node itself for which we want to create the ego centric network and its one hop neighbors.

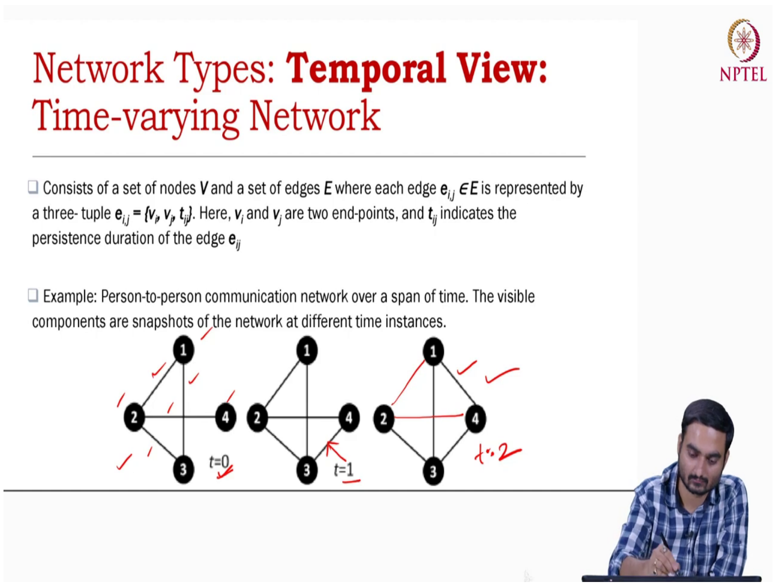
We only look at the one hop neighbor. So, for node A, the immediate one hop neighbor is are basically B, C, D and E, right. So, you have A, B, C, D and E ok. So, these are the nodes which basically form the egocentric network. Now, the edges are basically those edges which are connecting these nodes in the original graph for example, these edges this edge exists this edge also exists; this one, this one, this one, this one and this one. Now, if you see this you know this structure, right.

I have also connected those nodes which are neighbors of a right say for example, B, I have connected B and C, because B and C are connected in the original graph. I have connected C and D because they are connected in the original graph and so on. So, this is called induced sub graph; remember this term induced sub graph ok. So, induced sub graph basically consist of nodes.

So, I mean in the context of egocentric network I mean if we define, what is egocentric network, you basically say that egocentric network consist of egocentric network of a node u right consists of u its one hop neighbors and the induced sub graph of those nodes right, meaning that we will essentially pick those edges which connect these nodes, right.

The first hop neighbors and the node u itself from the original network and create another new small network. Now, this is called egocentric network. Now this node for which we create the network is called ego. Now, this is ego you see here this node V. So, V is ego and its one hop neighbors are called alters ok. These term terms are very important ego and alters induced sub graph ok.

(Refer Slide Time: 16:51)



Network Types: **Temporal View:** Time-varying Network

- Consists of a set of nodes V and a set of edges E where each edge $e_{i,j} \in E$ is represented by a three- tuple $e_{i,j} = (v_i, v_j, t_{ij})$. Here, v_i and v_j are two end-points, and t_{ij} indicates the persistence duration of the edge e_{ij}
- Example: Person-to-person communication network over a span of time. The visible components are snapshots of the network at different time instances.

Now, let us move to the temporal view of a network. So, what I mentioned is that you know sometimes when we scrape data, we also scrape the time information when a particular you know particular edge was created for example, ok. So, if the time information is not available or you want to ignore the time information then it becomes a static network ok, but if you want to incorporate the time information then it becomes a temporal network ok.

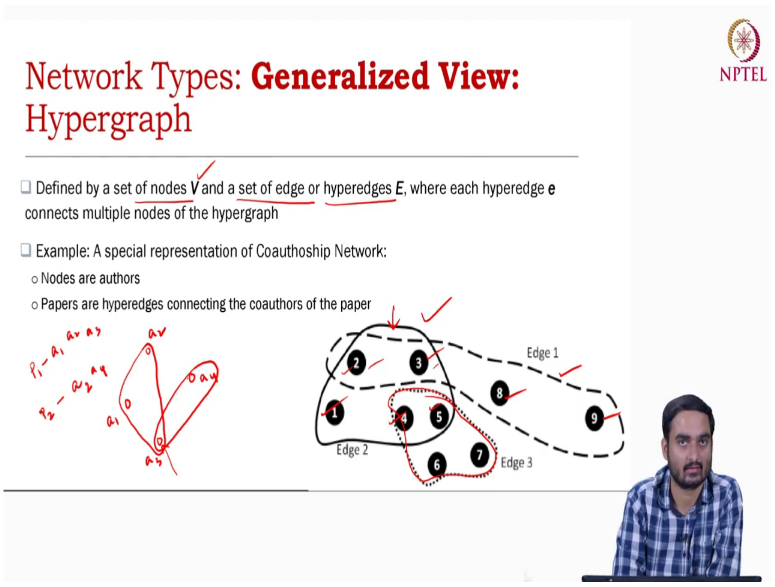
So, at every time you have a snapshot of the network for example, in this case in this figure you see a time stamp t goes to 0, there are 4 nodes and these are the connections right between nodes. At time stamp t equals to 1 you see that you know the network is almost same except the addition of this new edge, right. Time stamp t equals to 2 right you see that this edge has been added, this edge has been removed and this edge has also been removed, right.

Now, you can think you can actually think of all social network as a temporal network because say in case of Facebook nodes are getting added every minutes every seconds. Nodes are basically users. So, new users are getting added every minute new edges are getting added in the form of friendships, new friendships are getting created every second right nodes are getting deleted edges are getting deleted.

So, this is basically a temporal network the problem in temporal network not a problem, but you know essentially it is very difficult to keep track of you know the temporal the temporal dimension of every entity, right. You are basically saying that you know what a particular entity is doing over time and you have billions of such entities.

And it is very difficult to keep track of what that entity is doing right, but if you somehow if you are able to keep track of the temporal dimension, then actually you are done; you are essentially you should be able to mimic each and every activity of a social network ok. We will discuss in one of the chapters that how you can use temporal network for different applications ok.

(Refer Slide Time: 19:08)



Network Types: **Generalized View:** Hypergraph

- Defined by a set of nodes *V* and a set of edge or hyperedges *E*, where each hyperedge *e* connects multiple nodes of the hypergraph
- Example: A special representation of Coauthoship Network:
  - Nodes are authors
  - Papers are hyperedges connecting the coauthors of the paper

Now, this is the generic view of a network ok this is called Hypergraph. Now, what is hypergraph? So, in case of hypergraph right, we have again set of nodes indicated by V and we have set of edges. Now, these edges are not normal edges; these edges are called hyper edges ok. What are hyper edges? Hyper edges are basically edges which are formed using multiple nodes. So, far we have seen that an edge is formed based on two nodes.

So, two nodes are connected through one edge right, but hyper edge is actually formed based on multiple nodes ok. If you look at this example here. So, this is one hyper edge right and this hyper edge is basically formed based on node 2, 1, 3, 4 and 5. This is another hyper edge which is formed based on 8 based on 8, 9, 3 and 2; these are nodes, right.

This is another hyper edge which are which is basically formed based on 4, 5, 6, 7. You can imagine hyper edges as planes right, but you know we are not looking at the geometry as such. So, let us take an example of a hyper graph or hyper edge, right. Let us say the collaboration network again right where nodes are researchers and edges are not collaborations, but edges I mean edges are not simple collaborations, but edges is formed edges are formed in a different manner.

So, let us say you have paper 1 and paper 2; scientific paper 1, scientific paper 2. Paper 1 is written by author A 1, author A 2, author A 3. And paper P 2 is written by author A 2 and author A 4, right. So, this is A 1, A 2, right; A 1, A 2, A 3 right and say A 4 ok. So, the first hyper edge is basically this one ok. You can think of a paper corresponding to a an hyper edge, right.

So, this is a hyper edge and this is another hyper edge ok. And you see that these two hyper edges are adjacent because of this common node A 3 ok. Now, you can you are basically moving from a specialized representation of a graph to a generalized representation where you are not, you are not bounded you know to make a simple network where edges can only be connected I mean one edge can only connect to nodes.

You are basically making it generic where you are say that no edges can be formed you know by multiple nodes, right and so on and. So, forth you are basically creating a different notion of a graph, right.

Now, this is called hyper graph ok. So, this you know brings us to the end of the different types of networks, right. Later we will discuss you know the other types, other applications and other examples of networks that can further motivate you you know to study social network analysis.

Thank you.