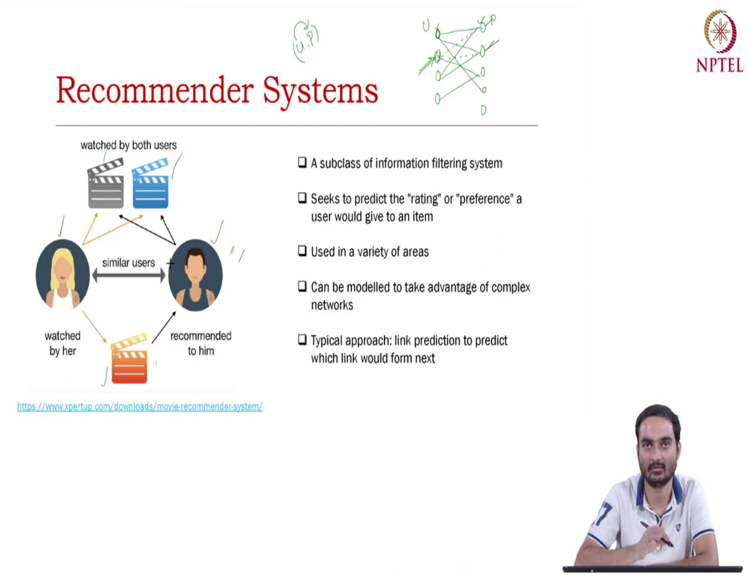
Social Network Analysis
Prof. Tanmoy Chakraborty
Department of Computer Science and Engineering
Indraprastha Institute of Information Technology, Delhi

Chapter - 10
Lecture - 05

So, today we will start another application which is the recommended system and this is quite popular not only for not only in the context of graphs, but also in general right.

(Refer Slide Time: 00:39)



And I think I have mentioned this thing multiple times that recommended systems are useful for E-commerce services like Amazon, Flipkart where products are being recommended to users, it is also useful for say movie recommendations right or say restaurant recommendation and so on and so forth.

So, what is the task? If you think of this task carefully, this is essentially a link prediction task ok. We discussed in chapter 6 what is link prediction? So, any recommendation system right can be modeled in terms of a bipartite network where in one partition you have users, another partition you have you know either products or movie or restaurant right.

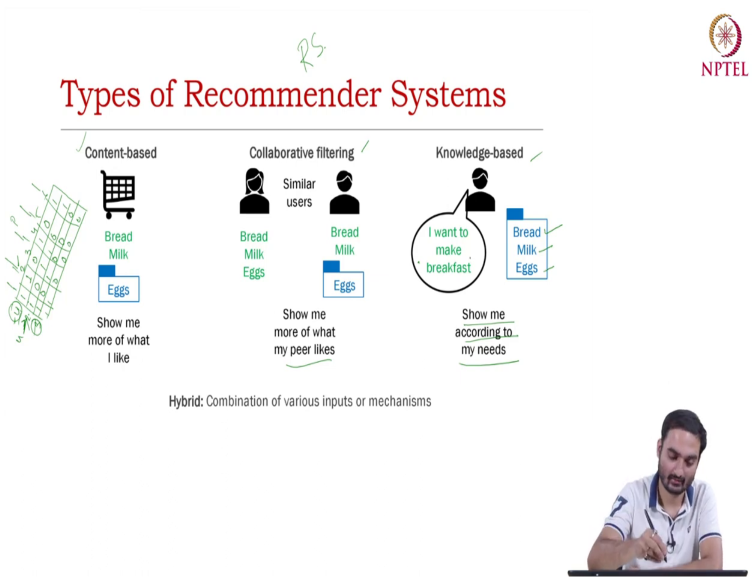
And if say let us say this is user and this is product and if this user has already purchased this product you can connect the user with the products right and so on and what is the task? Say let us say given this user what would be the recommend what would be the recommended products?

So, essentially say if this user has already bought right say this product and this product. So, that then these links have already been created. So, if you going back if you going in the past and if you going to the past and let us say in the past you had seen that this user had not purchased anything so far right. Now if you could correctly identify these two products for these users that would be your success right.

So, this is recommended system and it can be used for a variety of purposes not only for recommending products, but also for you know predicting the rating of a product right given a user product pair what would be the rating that this user is going to give to this product ok. This is this also comes under recommended system or recommendation system right.

So, as you see here a cartoon example there are two users both these users have already watched these two movies and they are quite similar and this lady has already watched this movie therefore, this is highly likely that this movie can also be recommended to this guy because these two users have kind of similar you know interests similar movie choices right.

(Refer Slide Time: 03:23)



So, recommend the system in short RS right its an extremely studied problem right its been there since last I think 15 20 years right in general if you think of approaches, the approaches can be broadly categorized into in three groups. So, first one is content based approach right.

Now you can think of I mean if you have historical data say user product purchase data, you can think of a matrix where rows are users and columns are products right.

Say this is user 1, this is 2, this is 3 product 1, product 2, product 3, 4, 5, 6 if this user has purchased this product you can write 1 otherwise 0. In fact, you can also write integers which indicates the number of purchases right. Let us say it is a binary right something like this right. So, in a content based recommendation system the idea is that the philosophy is that please show me those products which I like right.

So, in content based recommendation we look at the we look at individual content right we individual products and we also look at how this particular user has purchased other products and depending on that we recommend right we will not look at the purchase pattern of other users.

Whereas, if you think of collaborative filtering this is another approach here the philosophy is that show me more what my peer likes my friends like ok. Say for example, we already know that this user and this user these two users are friends right because say they have very similar purchase pattern and so on. So, if you and let us say user 1, user 2, user 3 are friends they are very similar purchase patterns.

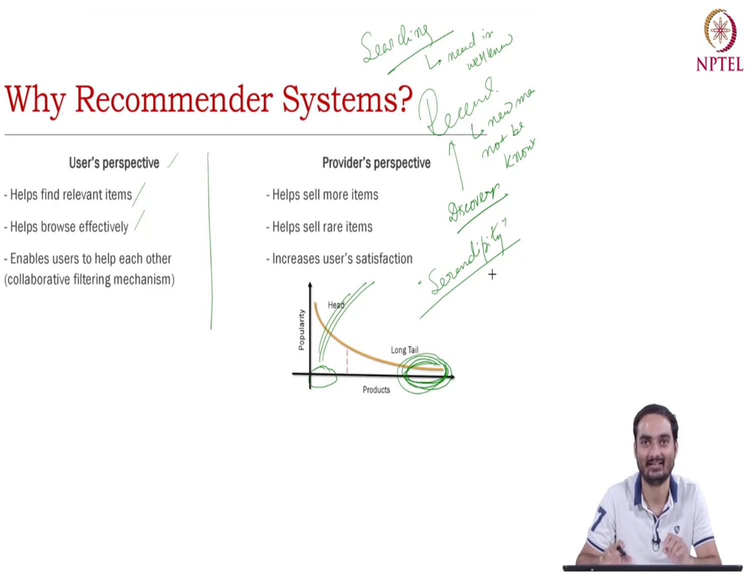
So, if you want to predict if you want to recommend products to user 2, you will look at the purchase patterns of users user 1 and 3 and depending on that you recommend product user 2 because the idea is that the recommendation depends on the behavior of a users neighbors right. So, it is very natural right. Say for example, say on Amazon right are seen in. In fact, movie recommendation say I am debating right on Netflix recommendation say when you want to watch a movie.

So, watch you want to watch a thriller movie right you can what we generally do? We generally talk to our friends who are close right quite close friends and we ask hey have you seen this movie? Have you watched this movie and if somebody says yes, I watched this movie you should also watch this.

So, I will definitely go and watch it right, but if I say I want to watch a movie and I ask my friend that have you watched this movie? And my friend says yes I have watched, but it is not that good it is ok. So, that would motivate me and that would possibly lead me not to watch that movie again ok. So, it is more of a collaboration behavior. The third one is more of a knowledge based recommendation. So, here the philosophy is, show me according to my need right.

Say for example, I want to make breakfast right I should be recommended say bread, milk, an egg because it should be known to the recommendation engine that I generally you know take bread, milk and egg in my breakfast therefore, there should be a knowledge graph or knowledge base based on which I should be recommended ok alright.

(Refer Slide Time: 08:10)



So, why recommended systems? In fact, I have already discussed from users perspective it basically helps find relevant items, it helps browse effectively it enables users to help each other right if you think of collaborative filtering from sailor's perspective. So, if you look at the product sale right you always see that there is a skewed distribution.

There are very few products right there are I mean there are few products which are I mean there are a lot of products which are less popular right and there are very few products; there are very few products which are you know a lot of which gain a lot of popularity. So, we generally see this kind of long tail distribution and the problem is to sell these kind of products. So, if you keep on recommending these products right ultimately your skewness will increase right.

So, there should be a balance there should be a balance of products which may not have been purchased by users in the past, but they are actually good right. So, those products can also be recommended ok. In fact, recommendation searching these are very relevant right as you can understand right. But there is a slight difference.

In searching the user knows exactly what he is looking for ok where is in case of recommendation user may not know what he is looking for right and when a new product is recommended to somebody. So, it may happen that the user does not have; user does not have any idea about that product, but when the product has been recommended, the user likes the product and he also recommends that product to his neighbors or friends right.

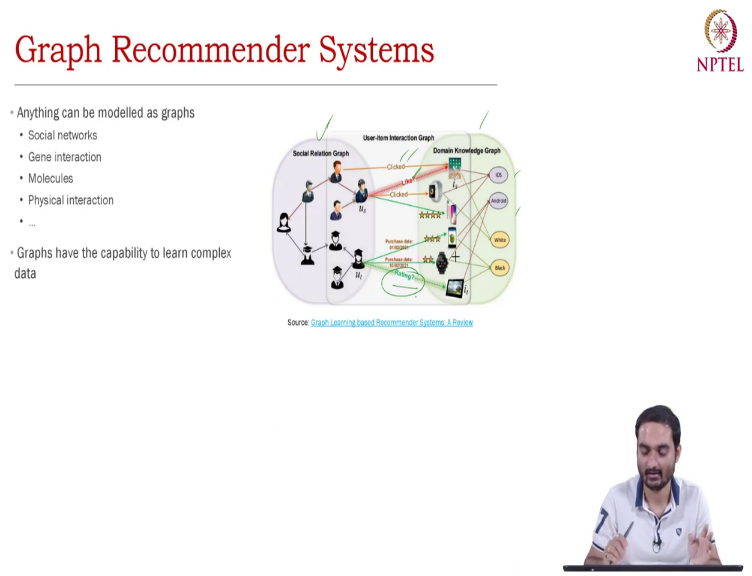
So, here the need is well known whereas, here the need may not be known right. Now if you are if you know the need say for example, you want to buy a Dell laptop. So, you type Dell laptop all the products will be recommended and you choose one of the products, but it may happen that while searching for Dell laptop you suddenly see a Lenovo laptop which is even more attractive and you end up you know buying the Lenovo laptop rather than buying the Dell laptop right

So, that is recommendation. So, it is also about discovery right it is also about discovery. Now when you talk about discovery there is another related term which oftens are which often is being referred to its called serendipity right serendipitous search. Serendipity is a phenomenon which basically says that when you are looking for something and you end up getting a completely different product or entity or search item which you liked right at the end of the day.

And this the end product has nothing to do with the such term that you know gave earlier right that is called serendipity. It is an it is a completely you know unorthodox discovery. Say for example, you are searching for you know yoga pose right some effective yoga pose and you end up getting a page we talks about cancer right and you start reading that page because after all we are all worried about this disease right and this is serendipity right.

Because you wanted to; you wanted to you know you wanted to find effective yoga poses say for your stomach, for your liver or for your say breathing issues right, but you end up getting a page which is completely different right, but you still get attracted to it and you start reading right. Well, I am not going into the details of all these things because this comes under information retrieval. But what I will do here I will specifically focus on you know graph learning for recommendation system ok.

(Refer Slide Time: 13:07)

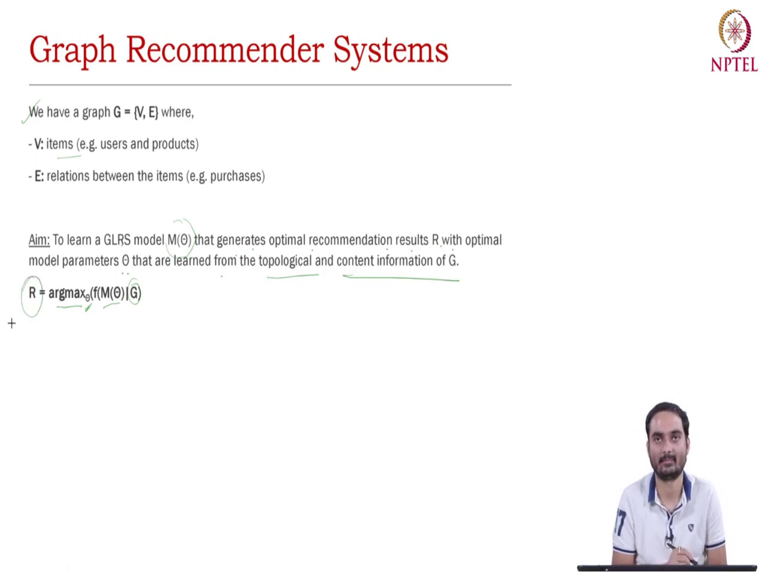


So, I mean this is at least at the at this stage you are you should already be convinced that why graph is important right. Graphs basically give you a perspective of you know this interrelations interdependency and so on. For example, you see a bipartite graph of users and products, but here users it is not a; it is not a normal bipartite graph.

Because here users are also connected in terms of some links it can be friendship links, it can be some other social relation whereas, products are also connected based on some categories some hierarchies right. And user group and product group they are connected through different links say clicks, likes, purchase and so on and so forth right. And in this kind of setting what is the task?

The task is to recommend a product to a user right or predict the rating of a user to a product right something like this which is unknown right. So, we will briefly talk about a few such methods which takes which take into account the user network, product network, use you know graph learning approaches the one that we discussed in the last lecture and you know effectively solve this problem.

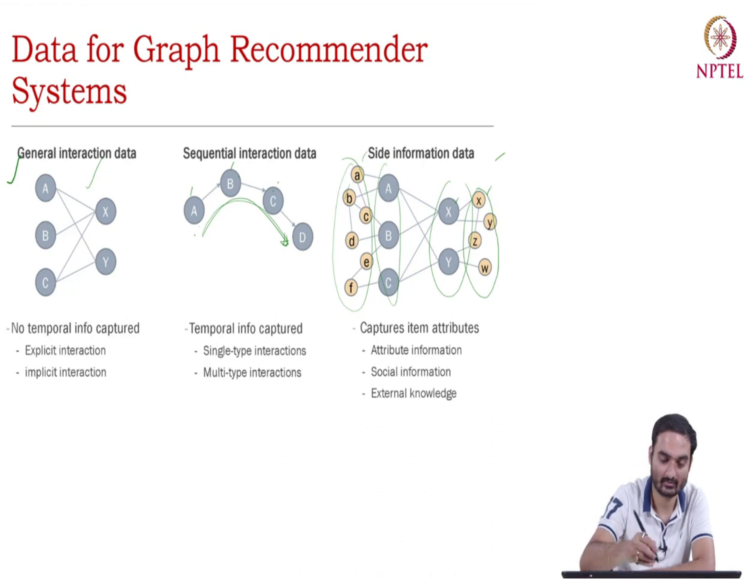
(Refer Slide Time: 14:47)



So, what is the problem statement here in graph recommendation system? The problem statement is we are we have a graph V comma E, V is a set of items this can be users or products and E is a set of relations say purchases right and what is the aim? The aim is to learn a graph learning for recommendation system model right something like M theta that generates optimal recommendation results are with optimal model parameters theta that are learned from the topological and content information.

So, you are essentially coming up with something which is a function which is parameters by theta which takes into account the graph structure and users behavior past history and it basically returns a rank list of products or items ok.

(Refer Slide Time: 15:58)

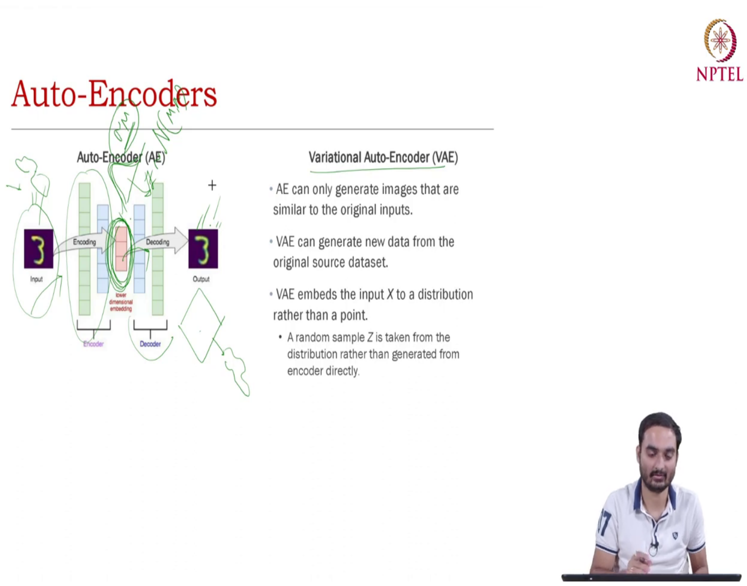


Let us look at you know the data that we generally use for graph based recommendation system. We generally model the data with a bipartite network as I mentioned earlier if there is explicit interactions, then it is good otherwise we use some sort of implicit interaction and model it in terms of bipartite network.

We also capture the sequential interaction data. For example, for a particular user if product if product A has been purchased before B, B has been purchased before C and so on then you can think of a sequence like this or for a product if A user A has purchased the product first and then B, then C and D you can think of a sequence like this right.

In fact, you can also think of a mix of both these things right it is a kind of side information data you have users, you have products users have different attributes for example, you know say location or date of purchase or you know bulk of purchase and so on. And for product also you have different attributes for example, the seller the delivery person right the time of delivery and so on and so forth. So, you can think of this additional meta information as attributes for products and users right.

(Refer Slide Time: 17:37)



So, let us look at the first approach. So, to understand the first approach we first need to understand another machine learning approach called auto encoder. Again, I am not going into the details of this comes under deep learning, but what an auto encoder does? Auto encoder basically is a neural network a deep neural network multilayers.

And you basically feed an input right you have some encoder which basically maps this high dimensional input to a low dimensional embedding and then we have a decoder which maps this low dimensional embedding to an high dimensional output right. So, and this output is same as the input right. So, what auto encoder does?

In case of auto encoder the input and output are same you basically try to come up with a low dimensional representation of this input right in an unsupervised manner because there is no supervision as such. Now this is auto encoder. There is another related concept which is called variational auto encoder VAE.

So, in auto encoder; auto encoder is not able to produce. So, auto encoder kind of stuff is generally used for so, for images right, but the you know the bad part about auto encoder is that it is not able to generate a new image right as you see the input and output are same right.

But if you use variational auto encoder it is able to generate a similar new image right say for example, you have one type of dog images image as input, you may be able to get another type of dog image as an output right and what auto encoder does? Again I am not going into the details of auto encoder, but what it is essentially does instead of learning the low dimensional embedding is basically learns some parameters right.

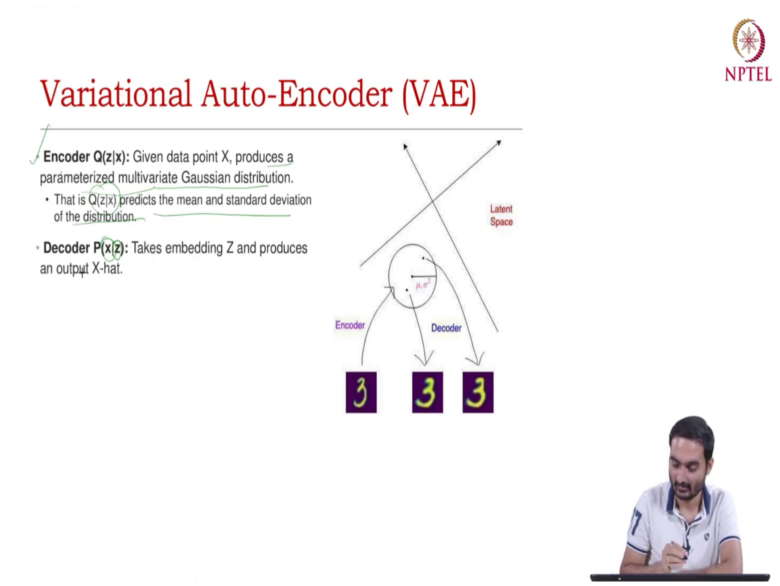
So, the idea is that you. So, the idea is that this latent space follows some distribution right and in VAE the assumption is that this distribution is basically you know a Gaussian distribution ok. So, in Gaussian distribution when you its multivariate right. So, when you characterize the Gaussian distribution what you need?

You need the mean and variance right. So, essentially instead of predicting low dimensional space, you are basically predicting distribution right. In other words, you are predicting the standard deviation and the mean and standard deviation right. So, when you predict the mean and standard deviation you are essentially you know you are essentially modeling a distribution right.

So, the encoder predicts the mean and variance right essentially a distribution and then from this distribution you essentially you randomly pick. Now this distribution is the distribution of this latent space right. So, whenever you pick anything any point from the distribution, you essentially pick an entity right. So, from this normal distribution you pick an entity right you basically pick an entity right and you feed it to the decoder and then decoder basically produces output ok.

So, the good part about variation auto encoder is that it can produce a different image a separate image which is which may not be same as the input right.

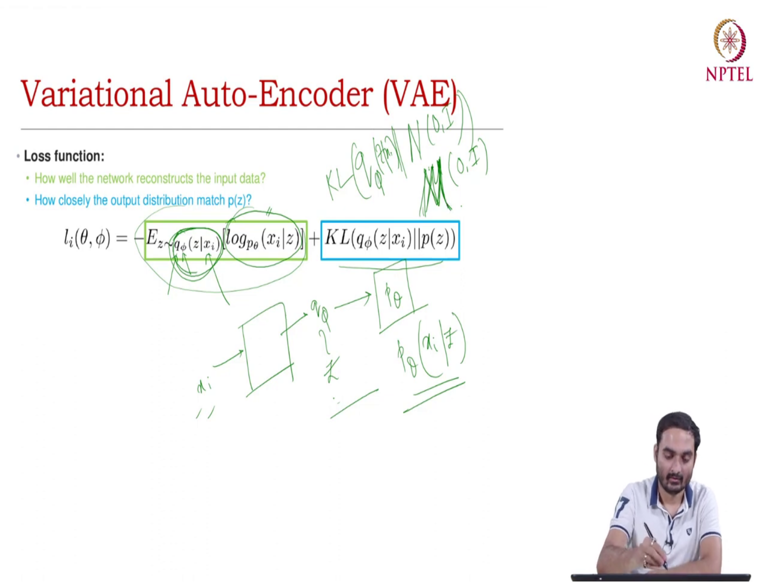
(Refer Slide Time: 21:39)



So, you see here there are encoder there is an encoder and there is a decoder encoder is represented by Q. So, Q basically takes an image X and produces a parameterized multivariate Gaussian distribution ok essentially mu and variance right mean and variance and then.

So, essentially the task of this Q is to predict the mean and standard deviation of the distribution and then you basically choose one item from the distribution feed it to the decoder right here Z is that you know the item that you sample from the distribution. And then you essentially predict the same I mean the same object right. So, this is the idea.

(Refer Slide Time: 22:38)

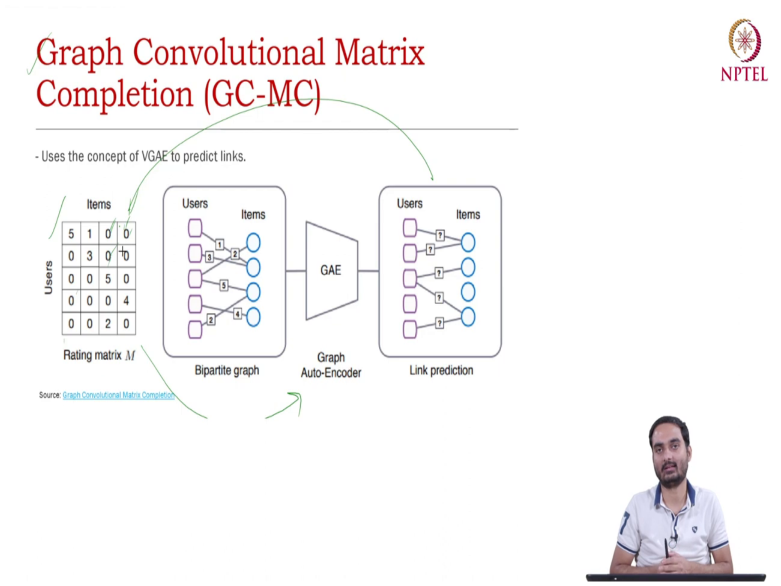


So, if you look at the loss function variational auto encoded there are two loss functions the one is the simple log loss right what it basically does? You see here. So, this is. So, Q is the encoder right this is phi is the parameter. So, and this is this takes x i. So, you have encoded x i is being fed here it produces q phi right.

Now, from q phi you basically randomly sample an item z right and this z is now being fed into a decoder which is basically p of theta right it p of theta it takes this. So, given this it tries to predict xi which is the input right. So, we basically measure the log likelihood right. The log likelihood ok of this output ok this is the first loss term and the second loss term is basically as I mentioned it aims.

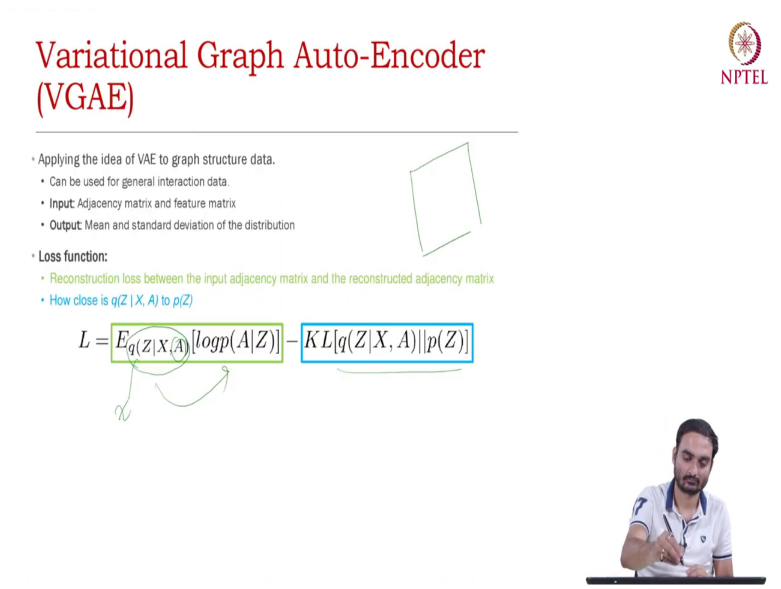
So, VAE aims to come up with a distribution which is Gaussian distribution right which is basically which is a Gaussian distribution right N Gaussian distribution. So, it tries to see how good is the distribution that you obtained from the encoder and how it matches with the Gaussian distribution right. So, it is basically a K L divergence between the you know the Gaussian distribution mean 0 standard deviation 1 and then ok. Now the same idea is has been used for recommended system using graphs.

(Refer Slide Time: 24:51)



Now this is called graph convolutional matrix completion problem you can think of this item user item matrix right and what you do you. So, you basically pass this matrix through a GAE right graph auto encoder types model right. And then you basically try to predict the you know the missing items for example, this item. So, it what it does? It basically reconstructs this one right.

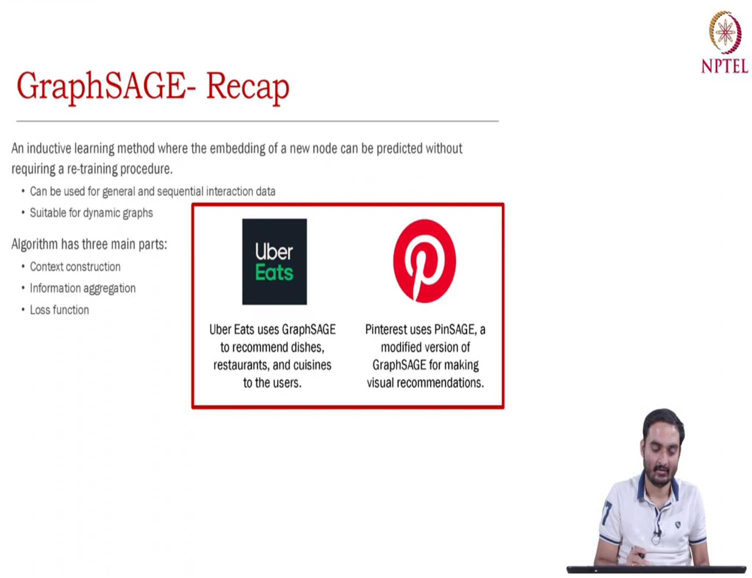
(Refer Slide Time: 25:37)



If you look at the loss function right its very same as the one that you mentioned earlier here the input is your A, A is at the adjacency matrix or whatever this matrix right user item matrix and then you basically you know you come up with a distribution right and then you randomly sample an item right feed it to the decoder measure the log likelihood and then you have a K L divergence side by side right.

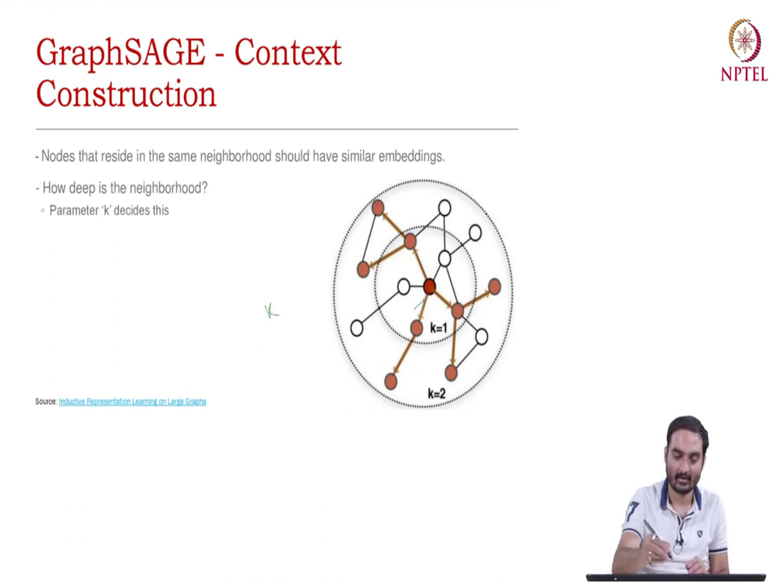
So, at the end of the day you will essentially get I mean a modified version of this matrix is user item matrix and basically and basically those items those entities which was 0 earlier right it can have some numeric some integers right here and then you use this integers right you use this new predictions for basically recommendations right.

(Refer Slide Time: 26:45)



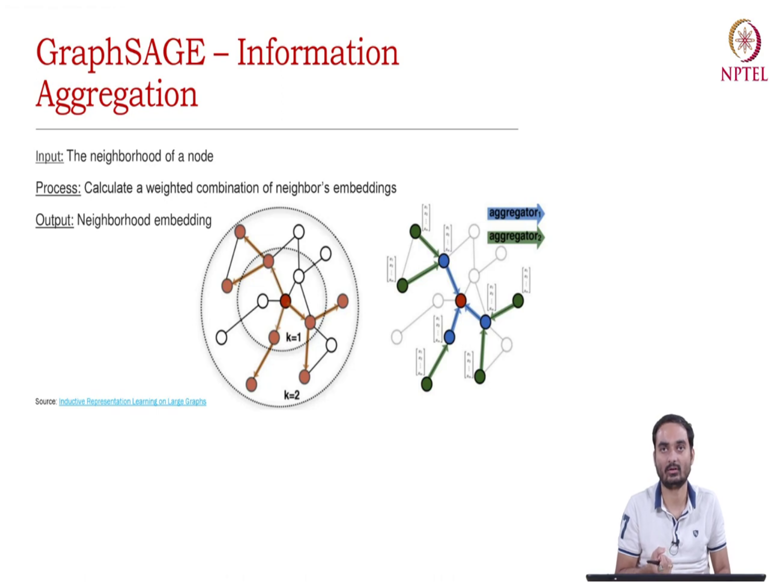
So, there is another you know graph based recommendation system which basically uses the graph sage algorithm that we discussed earlier. So, what is graph sage? Basically it is an inductive learning graph learning approach we discussed in the last lecture last chapter.

(Refer Slide Time: 27:04)



And what it does? It basically you know it basically for every node it looks at the neighbors you can look at k hop neighbors k goes to 1, k goes to 2 and so on. So, you decide this parameter k right and you essentially come up with kind of a subgraph of a particular node.

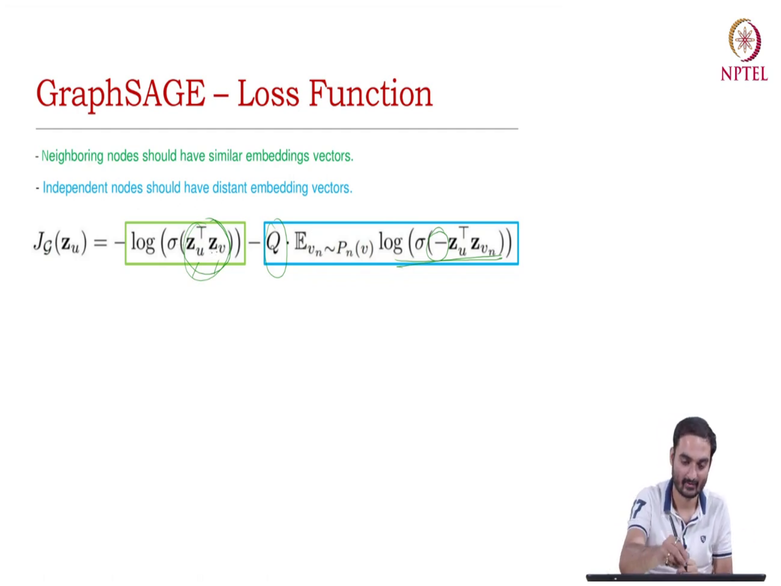
(Refer Slide Time: 27:28)



And then what you do? You now that that subgraph is your input right and what you do? So, the features of the neighbors are basically aggregated using some aggregated function and then you do some convolution operation and then you pass it to the next node and so on and so forth.

So, it is a kind of a generic approach I mean which is which turned out to be better than say GCN other graph convolution network type approaches and the good part about this is that it is inductive for a given for an for a new you know new node, it should be able to; it should be able to produce the representation.

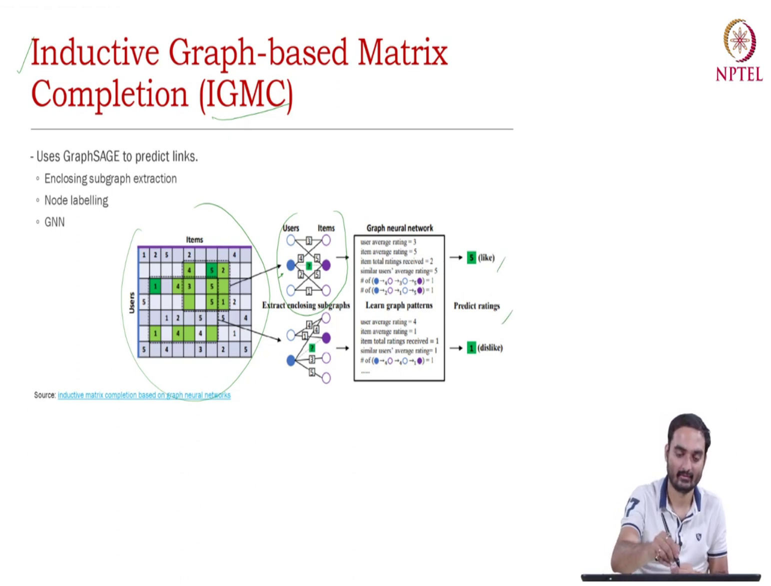
(Refer Slide Time: 28:11)



So, if you look at the graph sage loss function, it essentially maximizes the similarity between neighbors right let us say u and v are neighbors this is the dot product. So, you maximize the similarity between neighbors and you minimize the similarity between nodes which are not neighbors right you see here right. So, essentially for non neighbors you do you do negative sampling, you sample out non neighbors you take the dot product, you try to; you try to; you try to maximize it.

You basically try to maximize the similar minimize the similarity maximize the distance and you try to minimize the distance or maximize the similarity for neighbors right for neighboring nodes.

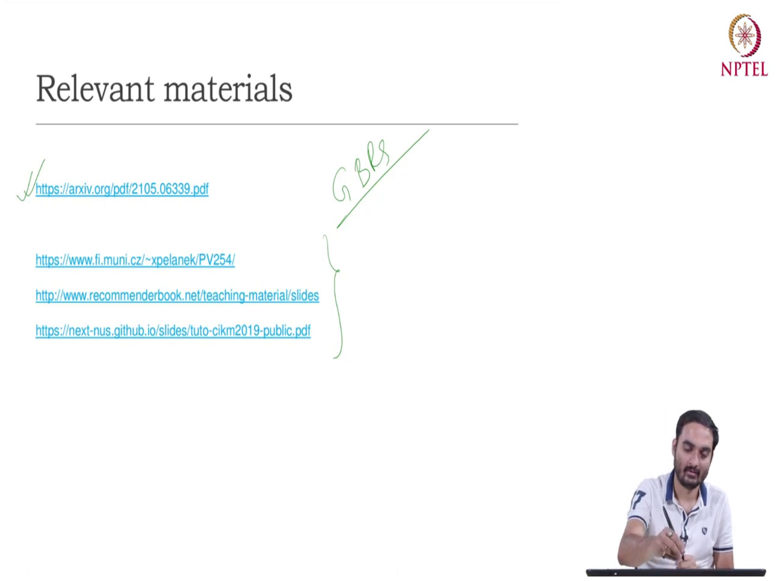
(Refer Slide Time: 29:05)



The same idea has been used for recommendation system in this paper this is Inductive Graph based Matrix Completion IGMC again I am not going to details of this. But the idea is that you have this user item matrix and from this you basically for a given user right you basically come up with an enclosing subgraph.

Enclosing subgraph is basically a nearby subgraph around that particular node and then you run GN based approach you can use graph sage kind of approaches and then when a and then you get the representation whenever a new node comes in you can easily use the pretend graph stage and then comes up with representations and then predict the likes or ratings or you know directly recommend products.

(Refer Slide Time: 30:07)



So, if you want to know more about this more about recommendation system, this is the survey paper that you should look at. These are the some other material that you should also look at if you want to know more about it more of more about graph based recommendation systems and graph based the graph based equipment system GBRs.

This is very popular these days because of the you know the effective usage of graphs right that you can basically leverage in the recommendation system. That is brings us to the end of this chapter. In fact, that bring brings us to the end of this course ok. So, in the next lecture we will formally conclude this course with major takeaways and some interesting discussions ok.

Thank you very much.