So those in such cases you do not have to worry about this you know meaningless attribute values being generated but you still need a distance measure so you will have to come up with some kind of distance measure which categorical attribute so if you use 1 out n or 1 hot encoding or anything else you still use some kind of a Jaccard similarity right so then do that okay.

Right so going back let us finish up the simple k means algorithm right so what you do with k means is so once you have done the assignment to the centroids if you forget the centroids estimate new centroids and keep repeating this until you no longer make any changes right sorry done?

No, no labels is necessary, there are no labels, there are unstable is learning from right. So if you have labels and other things, I mean depending on what your application is you might want to look at labels right, that is what the scameans is concerned, the question is the following. And you start with initial guess for the centroids right, I assign data points to the closest centroid right, recompute the centroids, then reassign the data points the closest centroids, I keep doing this until the centroids no longer change.

The question is have you done? Consider a, yeah, okay. Al right, how many clusters you want me to put them into? Three clusters I am done, you are getting close. So the probability of recycling is really small yeah, yeah whether other things which I can show is slightly more dramatic right, okay. These are my two sets of data points. Obviously, they are in two clusters, arent they?

These are the two initial centroids I start off with okay. So what are the two clusters I will get? Okay, and I compute the centroids for those, where will I end up with, and I reassign the data points to the centroids, I will end up with the same thing right so because of my bad initial choice for centroids, I do end up at a point where the centroids do not change anymore, but it is a really, really bad clusters.

Yeah, randomly chose those two points right, I randomly chose things. Exactly, so my question to you was if the clustering does not change or you are done, the answer is no okay. In this case surprisingly one of the few cases where the answer is not it depends, the answer is no okay. Because if you just do it once right, you are not done, so that is the thing with K-means in, you will have to repeat it multiple times right.

And please every time you said different random seed for choosing your initial starting point right. If you just the same random seat then you will get the same thing right. Yeah, so we will come to that right. And right, so the people understand why you have to do this multiple times right, because you will get stuck in some kind of local optima right, and you have to start over again with a different random set of K centroids right.

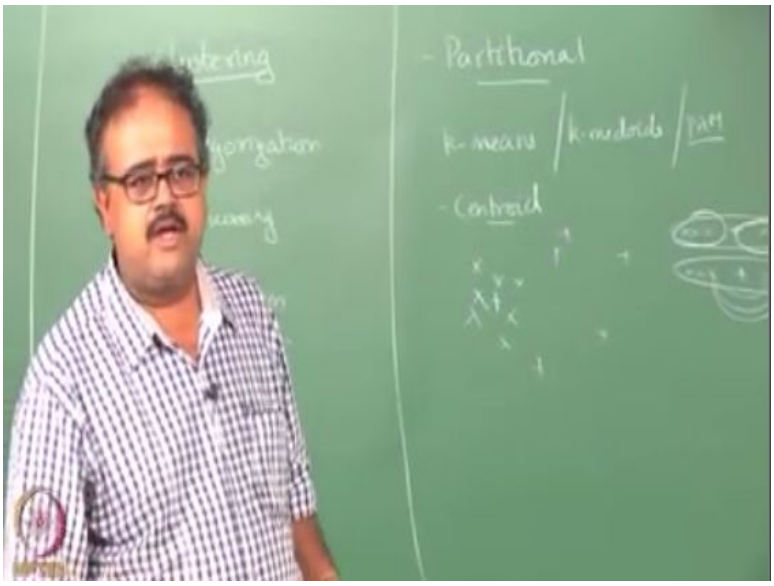
And then keep doing this yeah. Good point, so there are different measures right, I never told you about cluster evaluation measures so far right. So there are different ways in which you can evaluate clusters right. So more of the more popular is this some kind of a dispersion measure right, so I took a diameter, so the way the diameter is different there are two definitions unfortunately for diameter the literature.

So the first one is the average pairwise distance between the data points that belong to each cluster right so if you took that right, so I will take the pairs, pairs of data points right, I will measure the distance between the pairs. So but, then I will also have to consider pairs like that right. So I will look at the distance between every possible pair of points that belong to the same cluster right, and I will take the average okay, that will be the diameter of this cluster.

And the average of the diameters for all the clusters will be the quality of the overall clustering okay some kind of a dispersion measure, right so how much spread out my data point this right alternatively if you have if we are completing centroids right we can also compute the average distance of the data point to the centroid right and take the average and cross all the clusters.

And use that as your quality measure side so the either one is referred to as diameter in the literature so some call the and the most popular one is the average pair ways distance okay, is it fine right and I take it back so the average distance to the centriod is called the radius of these average is the centriod is called the radius of the cluster right.

(Refer Slide Time: 26:00)



So the diameter the second definition of diameter is the max of the pairwise distances right either to the average of the pairwise distances right or the max of the pairwise distances both could be call the diameter but the radius is essentially the average of the distances to the centroid, okay so this is one measure right so there is an another measure which I could think of which is essentially is called purity which did you guys look at purity already, will never gave it your in the first then assignments of array.

So purity essentially tells you for each cluster that you have what fraction of the data points belong to a the same belong to one class what is the largest fraction of data points and belong to

one class, okay so purity can be used only when you have data sets that have classed label associated with that right so suppose I have 10 data points I am going to one cluster six of them are in class one two or into class two or in class three, so the purity of the cluster is 0.6 right because out of 10 or in class one two out of 10 are in class 2 so 6 out of 10 is largest thing.

So 0.6 is the purity of the cluster so like this so whatever right so which are the level of the hierarchical or evaluating it at we will have some set of clusters, right we just evaluate the purity if you purity is a measure that it choose right so there is matter so purity mixing right, so something related to purity again if you using label data sets we can use something like entropy.

Right and what would be the entropy when the class distributed right look At the class distribution in the cluster so the P 1 fraction in the class 1 and the P 2 fraction in class 2 so that we can do the -P1log(P1) so that is entropy so that their classes are more are less evenly distributed then the entropy gives you better measure than the purity so the purity measure is the 0 .5 it is not clear to be weather the other 0.5 belong to the other class or the belong to the many other classes.

So that is the thing so these are the measure that reduces and then we having the zillion this is the satiation will be the clustering is the ill posed problem because there are many different in the clustering and then trying to list them in the popular one okay and the other one is called as the rand index which is typically used when you have the reference clustering and we have the reference clustering and I am trying to achieve the reference clustering so what is the rand index is the for the good point I don't have class labels.

So have the reference clustering but I do not have really but I do not have the reference clustering so the new comes in to the I do not have the class label so in the data points to the classes so the new set of the data and the new data point would not come I will learn new set of the data and I should run this algorithm in the in the that set of the data and then produce the clustering right.

So that the point here is I am not really looking to the reproduce exactly that cluster right what I want to do in that is the following I give you data point and run a cluster algorithm so I would like in the cluster on the training data it look like this so give me a new set of the data points to learn the clustering algorithm on it I will get the set of the cluster and then I little bit more comfortable with the set of the cluster and then the whole data point.

And then manage the reproduce the reference cluster a new set of the data points I get like actually get map to the whole data points itself so itself it slightly different problem so I am just evaluating the cluster algorithm and the clustering algorithm not the simple algorithm is produce so then we have the ram index and the one instants could be that we have nicely separated classes but many of them.

So that is the other case I am not talking about it so thermal index is typically used to every paie points in our data set so the F I and the F J belong to the same cluster in the reference clustering and then belong to the same clustering the cluster that they produce give the score of the one if the F I and the F j are the reference cluster and the difference cluster and the cluster that they produce and then give the score of the one and then how many pairs are there and Nc2 the pairs of them and then divide the some by Nc2.

And the what fraction of the pairs of the data points have you cluster correctly right what is the nice thing about rand index is suppose these are the more thing in the clustering that the original clustering is given to you right and then the end of the slitting points and then splitting in the two right the original clustering and the larger cluster and the original cluster and the original cluster.

And then this is the original cluster. and the rand index is not suffer a lot and the suffer greatly it is the only there cross and the cross cluster pairs that in the penalize within the cluster parts all fine so it gives you little bit in the terms of the lens in the terms of being in there stricter than the

other than the other and the reference clustering okay these are the different measure and we can use whatever you want right because it is not the rand index and then we can use diameter or radius in the case.

Right if we have the label data points and the we can use the label purity or the entropy or the full measure of these okay so we can list about the hundred of these measure and the people are used in the literature so just pick the favorite one yeah good point so how do you know which cluster is what? right so I have to figure and then I have to the alignment on the cluster right see I have some K1 cluster I have to use that is my reference cluster and my reference has k1 cluster and I have to use K cluster so which how do I align this k to the k1, right. So rand index gets rid of alignment problem okay. Oh okay no clusters are- yeah. Yeah there are minimum phase in which you can do this optimization right.

There are literally 100's of paper out there explaining to this kind of optimization so the question is yeah what is overhead and implementing this optimization is that whether in your particular application whether the overhead is justified because you are getting a significant improvement over the usual we have doing things right. So what people do as standard implementation or things which kind of give you good results across a variety of domains?

So if there is something very specific for your domain then you have to welcome to try to this optimization right and that is what make this engineering discipline as supposed to I supposed to theory right, so yeah a many things if you can do so went back right so now I did this I do tis again I keep doing this multiple times and what do I do with all the clustering I produced, take the best yeah that is right.

So is not nothing so you just keep doing the clustering and what are my evaluation measure you have so you take the best according to that evaluation measure, okay. So we threw away the rest so you have repeat it a few times and then take the rest, great. Okay let me re-writing it okay you

have to repeat k means multiple times do not please do not come to us after doing k means once and say that Oh it does not seem to work okay.

So I will guarantee you that one time it would not work okay and we do not know but just making that in fact emphatic. So let us go back to the question, how do you fix K? Okay that is one answer what is your answer, domain knowledge that is yeah exactly I mean depending on use of requirement domain knowledge right. So domain knowledge is one way of fixing K right is there are other more systematic way of fixing K.

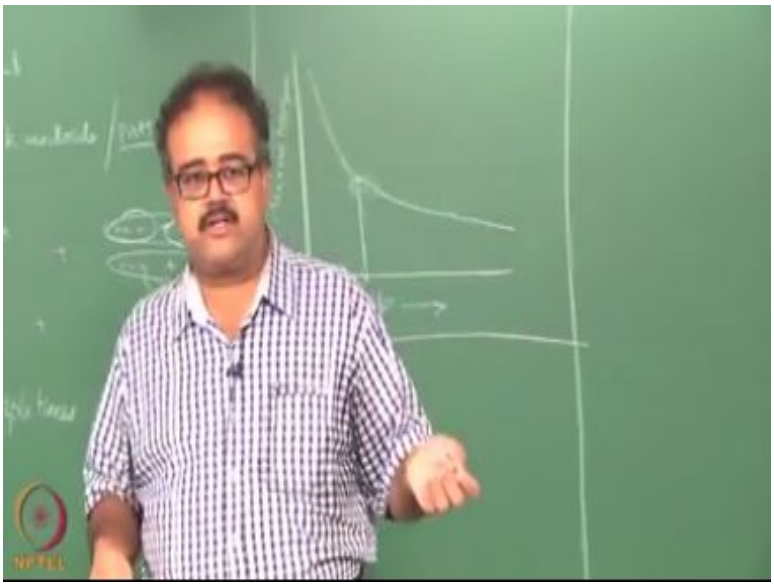
Try all ks but then how do I know which k is good so I tell you one thing say suppose I am using diameter as my measure, right. Larger the case and better it is exactly so the right way of doing it would be to say that I am going to have some kind of a complexity performance trade off right but it is incredibly hard to implement in K means right incredibly hard to implement in k means.

So but you can do that I mean people have come up with that so especially if you are going to take a basin approach to cluster right if you take a bayesian approach to cluster right how will you implement the penalty for the size so the number of k the prior what you will do with the prior who said prior what I will do with the prior, how will you make it penalize larger K exactly, so reduce the prior probability for large K, right.

But the you do the searching the main problem with this is most of this optimization thing is for finding cluster work well if I fix K but if K becomes a parameter and optimization it becomes incredibly hard that is why I say it is hard to implement right if K becomes a parameter and optimization it becomes incredibly hard to solve the problem right for the fixing a K right you can think of K mean as a approximation to solve the fixed K assignment problem right.

Even if for a fixed K if I have to do something like K means to solve it if I am going to make K a parameter and try to solve the larger optimization problem and it becomes little tricky so suppose to that.

(Refer Slide Time: 39:18)



So what people do is practical thing is draw a curve between K and let us say draw a curve between K and diameter right, so what would you expect to see? As K becomes larger decrease we will see something like that we will keep going down and keep going down yeah you keep going down to 0 when depends on how many data points you have right so if you at K = n it will become 0 I do not have to do all of them I already told you what the right value of K is here I have shown you an example say criteria what is the thing it bring, so may k initially decreases you can see a rapid decrease in the diameter right, after some point what happens the thick slows down right, so if you think about it that is a significant change of slope somewhere around here, right.

So that essentially quantifies your good curve complexity verses performances trade of, so wherever there is this change of slope right, you pick that point and say that this is the right k .

Usually it is you do this when you are trying identify a small number of k right, if you your is very large right, then you are probably using clustering for some kind of pre-processing rather than as the final means okay, so if you, we said I remember right, I told you that there are different ways in which we use clustering.

So if your clustering is very, very rare the case very, very large right, you are probably doing it using it for some kind of pre-processing right, we which case the exact choice of k does not matter that much right, when you are trying to use it as a algebraic visualization tool even visualization choice of care does not matter that much some of you using it as a actual end in itself right, you want to do clustering, you want categorization in those cases typically you have k smaller, right.

But it can run k up to a few 100 that side, right and then you can find the right value of k like this right, but if determining the right value of k is of some crucial importance to you and k is very large then I would not recommend partional methods at all, right so I would recommend other approaches for doing the clustering okay. So this is called the bend now what did you say it is that name for this it is very descript and they must call the Knee method, so it bends right.

So it is called the Knee method for define k okay, so in fact you can use this for in other cases also where you want to do this kind of complexity verses performance trade off and the optimization problem because very hard if it is know in the k into the complexity parameter also if you throw into the optimization, optimization problem because very hard so can use this kind of a empirical method for determining the actual value, okay, good and we move on right.

So what about k medoids , it is like k means so you find the centroid right, and represented and it uses the k is closes to the centroid right, so the every point of time you have a representative which is in actual data point, right and then when you do the assignment you assign it to the medoids that is closes to you, okay not a very interesting thing.

But the advantage of this is when you move from average to median, mean to median what is the advantage, in statistics it does not get effected by outliers same thing here, so when you move from centroids to medoids okay, it becomes little bit more robust to outliers right. Suppose of all this is the data points right, and I try to cluster it right, so I will end up getting a centroid somewhere there right, then but the medoid might be this a slightly better you know not that much but it is better than having a centroid that is out there, okay.

So one important thing I forgot it in an application of clustering something call outlier mining, so what is outlier remaining that find outlier right, I mean that is nothing how big deal about it, so why do you want to find outliers, delete term is one of it anything else sorry, fraud detection right, so I want to do any kind of anomaly detection so outliers would be anomalous data points, so who said fraud detection show me yeah okay, yeah.

Yeah so the outliers would be some kind of anomalous data points and therefore you would want to find them I am not interested in deleting them from the data set where I am probably interested in deleting them from the real world right, so that I want to catch these things and put them safe guard against them and things like that. It will be useful for understanding yeah; it is the one of the initial thing I told you right. so instead of randomly sampling around the entire state space right generating 10000 samples from the million sample data base to clustering with k = 10000 and sample from each cluster, that gives you more samplings.

I mentioned that in one of the uses in very beginning okay great. So we have done k- means, we did k mean right okay, so PAM is called the partition around medoids right. In fact it is incredibly expensive algorithm, nobody uses PAM any more. When it was proposed it was very big thing and but then people came up with faster ways of doing PAM, all of these work on very small data sets okay. Really large you want to do 10million data points things like that. PAM is nowhere near competition, any of medoids or not at all competitive.

On a very large data sets and any way I will just talk about it because it was very interesting algorithm right. So in partition around medoids, so what you do is okay. So I have some data points, I start by assuming some, say I am doing some clusters, I will start by assuming some two dada points as my initial medoids right. So let us say that this as 1 medoids right and unfortunately assume that this is the other medoids right. Now I look at the quality of the clustering, let us say I use radius as a measure for the quality.

So I will assign all the data points close to right, end of this is one cluster and I look at the average distance of the data points to the medoids count keep that as my point to medoids, I will keep that as my quality of clustering, right now what I will do is for every medoids right, I will consider swapping it with the non medoids right. So I will say, I will make this a normal data point right that we make that a medoids. Now if I make that a medoids what is the change in the quality of the clustering?

Likewise for this medoids I will consider each non medoids intern and consider swapping with it, or whichever gives me the best improvement in clustering, I will keep that as the my new non medoids and then I will go look at the other medoid. Now I will consider swapping this with each one of the data points intern right and then I will be swap it here. Sorry anywhere, so this how it works, it is very expensive as I told you right. So for every time you do the swapping you do the order n thing.

I just have to check the distance, so checking the distance rather order in computation right, so essentially I end up doing n^2 computation for very swap, that I have to make. People made all kinds of interesting observations and they came up with the ways of cutting down on the number of the computation . So when I make a swap I do not have to go through each and every PAM, so only those change cluster membership. Only those data point change cluster membership I have to really look at it.

Data points belong to the current cluster right the medoid of that change; obviously I have to do re computation for that right. Among all the data points do not belong to the clusters, only those change clusters memberships, I have to evaluate this right. But then I have evaluated the cluster membership anyways okay. So I do not have to do new things but still I have evaluated the cluster membership. So but then you can again organized it little more efficiently so for.

So depending on how you organize this computation people come up with a variety of different things that is PAM I can remember, that is partition is on medoids right. People are interested I can give points to read out on more PAM things like that, like I said it is not very that widely used in the community so we will skip those things. So what is the problem with K- means PAM addresses? Initial random really does not matter anymore because I am any way considering everything; I am any way considering very possible pair gone.

Then well it is medoids it is no long that affected outliers right, what about the choice of K, we still have to choose k that is still there, it is not gone away right. Right what about the issue k-means if you have real value attributes, works well if you have attributes, if you gotten rid of that. I still need a distance measure right. So if I am going to have categorical attribute better have a distance measures that takes care of categorical attribute, still the close problem remain with PAM okay great.

IIT Madras Production

Funded by
Department of Higher Education
Ministry of Human Resource Development
Government of India

www.nptel.ac.in

NPTEL

NPTEL ONLINE CERTIFICATION COURSE

Introduction to Machine Learning

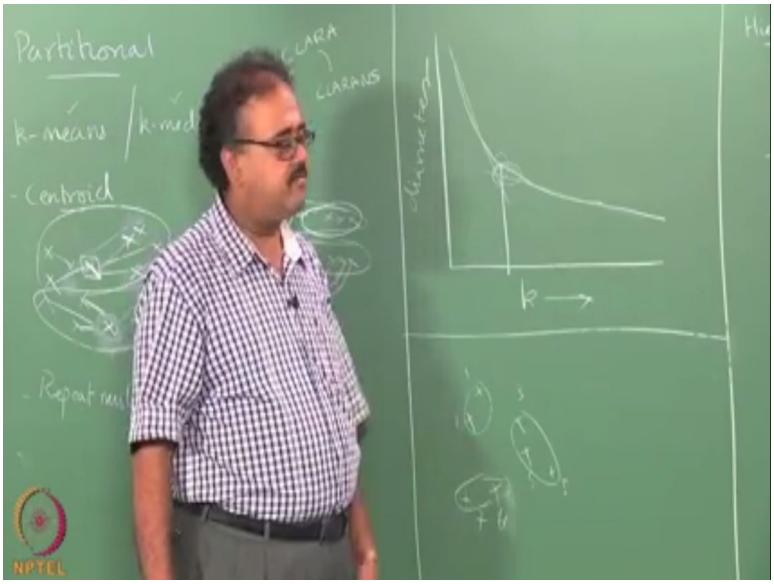
Lecture-71
Hierarchical Clustering

Prof. Balaraman Ravindran
Computer Science and Engineering
Indian Institute of Technology Madras

Hierarchical Clustering

Hierarchical Clustering.

(Refer Slide Time: 00:17)



So in hierarchical clustering what I will do is I will start off with there are many ways in which to do this , one way to do this is to say that I will start off with each data point being a cluster off its own .

Each data point is a cluster of its own and then what do I do I try to merge them into larger clusters .

this is not a special distribution of data points okay just put these things here that is I mean individual clusters data points are individual clusters then what I do is I compare distances between the data points and say maybe merge these merge these merge these merge these .

So when I draw a line like this it means these two have been merged then likewise these two are merged and then the third data point was merged with that and then these two got merged .

I should have one two three four five six seven one two three four five six .

So I initially start looking at this way okay one and two are close together let us merge them three and four are close together let us merge them five and six or six and seven are close together let us merge them after I have done these things okay next thing I look at okay five is close to three and four okay, let me merge them and so on so forth and next what we would want to merge?

That should bring you to the question how do you measure the distance between clusters you know how to measure the distance between data points and how do you measure distances between clusters

even how do you know that five is close to three and four

you could do a variety of things there are many different measures that you could use .

so centroid based sometimes how that another thing that people use is called single link distance you know what a single link distance is?

I look at two clusters. I look at the pairwise distance of taking one point from this cluster and another point from that cluster . I look at all possible pairs

I look at the closest to such a pair, look at the closest such pair and then I use that as my distance between the two clusters. What do you mean by not equal every possible pair?

there are five here to here I do ten pairs okay,

you could, what do you think that is called that is another decision we should call average link clustering okay the single link clustering single link clustering essentially takes the closest data points if we take the closest data points here which is closer I already merge this this is one cluster of now that is another cluster this is in the cluster which is closer these two are closer at least or closer the question now boils down to is one closer to three then four to six.

here is not very clear to me but I will take your word for it okay now that is basically done all the data points are merged at this point

I did four and six because there that a single link more I mean single link is by far the most popular hierarchical clustering distance measure and then there is another one called what do you think complete link would mean not summation, farthest max.

I look at pairwise distances and the max of these pairwise distances is assigned as the distance between the two clusters

in single link clustering distance between these two clusters was given by the distance between one and three and the distance between these two clusters was given by the distance between four and six that is single link clustering in complete link clustering the distance between these two will be given by two and five or one and five were okay.

Whatever two and five and the distance between these two will be given by seven and five , what you think is larger? Oh my god!

It is hard to make it how to make 2 and 5smaller is it?

I'll make it easier for you there, yeah two and five is moving two and five smaller than 7 and 5 if I am doing complete link clustering then I would have merged these first and then I would have merged .

I could do a centroid based distance equal to a single lane complete link I could do an average link anything else you can think of yeah, I could do radius based. What do I do? I will take the two clusters. I want to find the distance between these two clusters. I essentially merge the two clusters and find their radius. If I want to find the distance between these two clusters I will merge these two clusters and find their radius.

the smaller the radius the closer the two clusters are

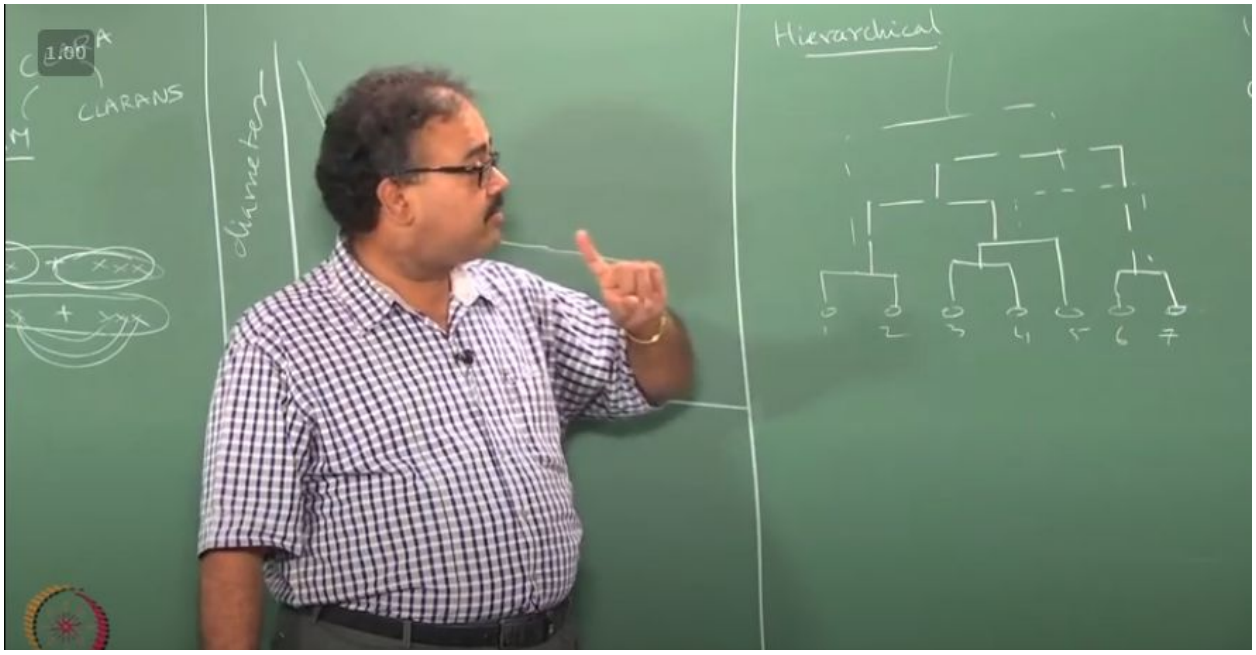
no centroid I am looking at the distance of the centroids of the two here, I am merging the two and then finding the centroid of the merged cluster.

it is different

I like and similarly I can do that you know diameter

I can merge the two clusters take the diameter, the smaller of the two is the better these are essentially more useful for comparison purposes I want to know whether cluster one is closer to cluster two or to cluster three then I can merge the two find the diameter and then make a decision it is not really a true distance measure in the sense that I cannot work by say what is a distance of cluster 1from cluster 2 okay diameter does not make sense .

But then if you want to say it is Cluster 1closer to cluster 2 then to cluster 3 then I can use the merge diameter and I can make those decisions okay, all of these are valid ways of doing hierarchical clustering yeah, you could I mean define whatever you want these are popular ones yeah and yeah they do use other distance measures as well for doing hierarchical clustering okay.



MEASURES:

1. **Single Link**
2. **Average Link**
3. **Complete Link**
4. **Radius**
5. **Diameter**

well now it is a nice thing about hierarchical clustering once I choose this distance measure , think one minute stop and think these are meta distance measures I still need to decide on point-wise distance measure .

that could be an euclidean measure when I say single-link I said that distance between the two closest points but what is that distance? That could be Euclidian that could be Jaccard similarity that could be whatever you want absolute deviation whatever then you can look at whatever distance measures you want.

that I still have another distance measure, that is dependent on the data type that distance measure typically depends on the data type, well this distance measure typically depends on the

kind of clustering that you are looking at . Once I had a tree like this, what you did not have to choose here? K I did not have to choose K here, once I have a tree like this how do I recover clusters if you think about it when you completed my clustering all I was left with was one cluster ?

How do I recover the cluster? Traverse the tree, how would you traverse the tree? Now when they start here at a single cluster if I come here I have individual data points yeah, I basically have to figure out me point okay I will break here if I break here how many clusters do I end up with three clusters if I break here how many clusters I end up with two okay I could choose to break at me point in between and then say that I will take that many clusters I get .

How do you choose which point to break it at?
I can do the knee method again . I can do a K versus evaluation and I can get this thing, what is the advantage of doing hierarchical clustering is I get that entire graph generated to me in one go .

I am not actually having to rerun everything for different choices of K I get the entire K graph that knee graphs are generated for me in one shot you do not, you know well yeah you do not as many as you can get see the point is the goals that you do not get or the ones where it was very hard for you to find a breaking point .

essentially if you choose a threshold at which you are merging their points , all these things get merged there is no real reason for you to choose three over seven or something if you don't get anything for five six in between after three you move to seven, maybe we did not make any difference to look at four five six usually that is what we will end up
great.
depending on what kind of a measure you choose you end up with different kinds of clusters for example if we choose a single link?

What does a single link say? That closed the distance between two clusters is the closest data points so I could have a cluster that like that another cluster that is like that and another cluster that is like this okay

okay we tell the two you should merge now?

where well give me names leave me numbers one two three. So for you single link clustering I will merge one and two if you single link I will merge one and two I will basically get this humongous very long cluster .

So that is the problem with single link clustering you might end up with very long clusters essentially the points at one end of the cluster to the other end might not be very related at all that is the problem with single link clustering yeah on the other hand if I had used complete link clustering I would have merged two and three first and then I would have merged it with one but if I use complete link clustering is highly unlikely that I would actually produce say such elongated clusters as one and two in the first place , single link clustering tends to produce very tight small clusters .

And at some point you then merge a lot of clusters but then you will merge them at very high levels in the tree. It is the lower levels in the tree you'll be getting smaller clusters. Okay ah where is the tree here? that thing that figure I drew there is a tree okay but it is not called a tree in the hierarchical cluster in literature it is called something else
Dendrogram, you know what the dendrogram means?

It's a tree yeah dendrogram means tree just said they went to a different language and pulled out the one loaded three okay, dendrogram is a tree . So that is a dendrogram and what are these levels I am talking about that yeah no not really iterations they are the levels at distances at which I merged .

So when I merged one done to , I had some threshold I start off with one I say anything that is within point one distance unit of it I will merge into a cluster there's nothing it stays as one and then I say okay within point two point three point four now at point four great, now I have two at the level of point four I have merged one and two and also turns out that level of point four emerged seven and six and three and four .

That is why all of this should be at the same level but you can think of this being slightly different because three and forests are slightly farther apart then one and two all , these levels are essentially the distances at which you are merging there . So that is why five gets merged with three and four at the higher level because if I slightly farther away from four then three is that is the reason for the levels and then these two get merged at a higher level because we are farther away, the levels in the tree at which you merge or essentially the distances rate at which you are doing the merging okay.

IIT Madras Production

Funded by
Department of Higher Education
Ministry of Human Resource Development
Government of India

www.nptel.ac.in

NPTEL

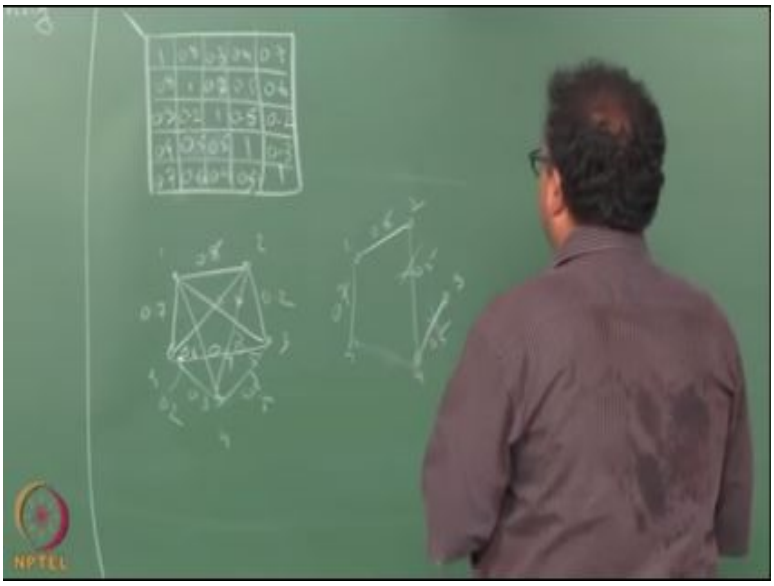
NPTEL ONLINE CERTIFICATION COURSE

Introduction to Machine Learning

Lecture-72
Threshold Graphs

Prof. Balaraman Ravindran
Computer Science and Engineering
Indian Institute of Technology Madras

(Refer Slide Time: 00:14)



When you looked at many such things so, so I am going to look at a specific setting now which essentially looks at how do you do clustering? Right, when I do not really give you the data points right, but I give you a similarity matrix between them right. So I do not give you the data points right, but I tell you that okay, here are the similarity like similarity matrix between the data points, so this is like say 0.8 similar is what are the things I can fill in now, I am trying to

give you something consistent when this cooking this numbers up on the fly. I cannot cook everything up on the fly can I? something like these rates.

I will give you a matrix like this okay can you do clustering? The similarity is the inverse of distance right or I can, of course, take the inverse of this and give you the distance between the points as well right, so I give you a similarity matrix right. So the reason I am stating this is sometimes it makes it really convenient to reduce the data that is given to you and even if i give you let us say i give you a huge collection of documents right, so instead of computing the distance between every document again and again when you are doing clustering right.

I can just basically do a n x n I can construct the n cross n matrix like this right. In fact, I am assuming this distance is symmetric right I am assuming the distance is symmetric so it is not really n cross n is only half of that right, and so you can construct this matrix you can keep this with you and then you can do clustering based on this right. Suppose I want to do something like k-means's, and how will it work in this case? Little tricky right I want to k-means is a little tricky sorry yeah but then the first you have to find embedding right.

So it is not, so that is a is called an embedding into a space right, first, you have to find them adding (A) be the embedding,

(B) might not be sufficient right may not be sufficiently accurate you have to, first figure out what dimensional space you are going to do this embedding in, this is 2d 3d, 4d and finding the embedding itself is a hard problem right, and then you want to do clustering on top of that. It is you are going to actually solve a harder problem before you are going to solve clustering. So you do not want to do the embedding right this has some other mechanism which you can do this right. So one way to think about given data like that is to think of it as a graph right, think of it as a graph and think of it as this some kind of weights between the nodes right.

So I have how many nodes I have five nodes right, so I have five notes right, so I will give them numbers right, so 1 to 2 okay the weight is 0.8 that is a complete graph by the way. This beginning then 1 to 3 right, the weight is and then also 2 to 3 and becomes more and more at this 0.5 belong to. Now I am confused 4 to 5 is. Okay really I mean you can make out the weights know right. So that is a graph right, and I want to look at a partitional clustering on this graph right. So what I can do is I can solve what is known as a min-cut problem on the graph right.

So what is the min-cut? A cut on a graph is a set of you can do to two things, you can either cut on the edges, and you can cut on the vertices? We will worry about cutting on the edges that cut set of edges on a graph is a set of edges such that if I remove the edges in the cut-set the graph gets split into two components right. So I take a connected graph I remove a set of edges from the graph, okay, and the graph becomes two separate components okay, so that is called the cut-set right and the min-cut is a set that has the least weight right. In an unweighted graph, it is a set that has fewer stages right.

In the weighted graph, it is a set that has the least weight it could have more edges, but of all the weight edges could have less weight then that becomes a min-cut okay. So you could try and do a min-cut on this graph right, so that is one way of solving it and so in the next clustering class that we will have which will be like not next week the week after right the next clustering class will have I will talk about spectral approaches to clustering right. Which essentially talks about different ways of solving this min-cut problem, talks about a completely different way of solving the min-cut problem, so we look at spectral clustering later right, there are a couple of other things that you can do right.

So especially all of you have done graph theory some point you must have done graph theory all of you have done some graph theory, basic graph theory data structures in okay. Do people understand what the meaning is by minimum spanning tree very one understands what a

minimum spanning tree? Is so what is a minimum spanning tree, a tree okay, bit spans all the vertices it connects all the vertices and three it has the least weight among all those trees that connects all the vertices right. These are the three things I so minimum spanning tree you can just basically take each term and define it and we get the think.

So in this case, if you can think of a minimum spanning tree, what would it be I am making people run the extra, or something now come on Kruskal prim what do you want to run Kruskal okay give me a minimum spanning tree now right. So that is a minimum spanning tree, so I started off by inserting the edges with 0.2 writes, and then I looked at things that are outside and figured out which is the least cost eight, so both of these had the same cost I so now I have a minimum spanning tree. Now I can once I have the minimum spanning tree I can use this to produce clusters. I can start off by saying, in this case, it is pretty trivial.

I could start off by saying remove the highest weighted edge in the spanning tree, remove the highest weighted edge in the spanning tree right or should it be the lowest fit we are doing similarities right by doing something okay remove the lowest weighted edge in the spanning tree now I could do this either way right if I had add distances instead of similarities also I could do this right, remove the lowest weighted edge. No wait now I think have to the other way around I am doing similarities right. So I should do a Max spanning tree may not have min spanning tree. This is it easy to do a Max spanning tree is the same complexity as a min spanning tree okay.

So you did not tell me what a Max spanning-tree here 5 is and 6 is 0.2, 5 and 4 right not is that yeah okay if you want to do it that way sure 5 to 3 is a 0.4, 5 to 3 is 0.2, so 5 to 4 is a 0.3, 2 to 4 the 0.5, yeah that could work yeah so that is a point okay right. So now what I do is I remove an edge that is got the least weight right, so I will remove this guy, so I am a left with two clusters it is an if i remove an edge from a tree it becomes disconnected right. So get the Max spanning tree in this case I remove the edge with the least cost right. So I can think of doing clustering by doing it this way right.

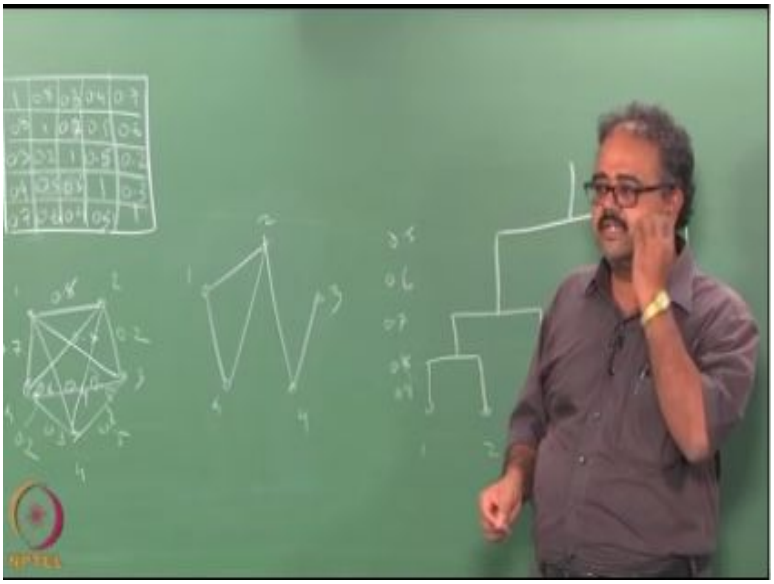
In stuff, if I had been given distances instead of similarity I've done the minimum spanning tree right removed the edge with the max cost right. Now I did the Max spanning tree and remove the edge with a min-cost to give me two clusters, and if i wanted to do work with distances instead of similarity I will do the minimum spanning tree and remove the edge with the max cost right okay this gives me 2 class suppose I want 3 clusters what do I do? Remove another edge, so now that two-three classes will be 125 will be one cluster and three will be one cluster by itself or will be another cluster by itself right.

So I do not really need to do the embedding right I can treat it as a graph right, and I can still do useful clustering with this. So one thing is to do the min-cut which we will come back to later other one is to first do the minimum or maximum spanning tree right depending on what data are given and then do this okay cool. I am going to look at something else I will erase that okay, so that is a good question. So take it to pick you to know so you have to use some other heuristic even here also there were two possible choices for my first stage itself right I could have when I wanted to cut a 0.5 that I could either cut the one between 3 & 4 are the one between 2 and 4.

That I chose to cut the one between two and four because it gave me more or less equal-sized clusters that could be here to strictly use right. So you can say that okay if I cut this 0.5 I get an isolated node and all the other nodes are in one cluster, look at the other 0.5 I get two nodes 2 and 3, so maybe that is a better division, so you can use additional heuristics like this there are multiple things that are possible. In fact, it is even more complex than that there could be many minimal minimum spanning tree is possible. I just showed you one tree it luckily it turns out that this particular graph there is only one minimum-maximum spanning tree that could be minimum many maximum spanning tree is possible. What do you do in that case?

You could only just pick one that is it just pick one and then go ahead and do the clustering it is like yeah there is no single answer for this remember me telling you clustering is an ill-defined problem yeah, so there is no single answer for this right there could be multiple different answers. So let us look more interesting, okay, so I am going to introduce you to this concept called threshold graphs right. So I have graphs like this what the maximum similarity I can have one right I will start off by saying I will connect all the nodes in the graph okay, says that the similarity is 1 or > 1 okay is.

(Refer Slide Time: 15:29)



So I will basically end up with that is my graph right, so it is essentially the empty set right now I will say that okay great I have this graph and I am going to treat all the connected components in this graph as a cluster. All the connected components in this graph I will treat as a cluster, so what do I get five clusters, so remains you of something hierarchical clustering this how we start off in the hierarchical clustering rates I will say that I will start off with each data point of being a cluster of its own right. Now what I do okay, I will start decreasing my threshold right so what the first step that I can do is? I will make my threshold 0.8.

Now I will do all my connected components right I will take them as clusters, so how many clusters I have 4 clusters 1 and 2 right. Next what is the best of sorry point it next I say oh 0.7 right that is my graph 0.7 that is my graph so what does it do, so I change the labels here if people are wondering what happen? So and then what is the next level I can do 0.6 is it and what is 0.6 is 2&5 is 0.6 okay sorry does not matter this is still connected right nothing changes I do not hit use any new or connect components it is the same set okay. No new clusters have been found right then I will go to 0.5 so what happens to the 0.5 I get that anything else is a little tricky.

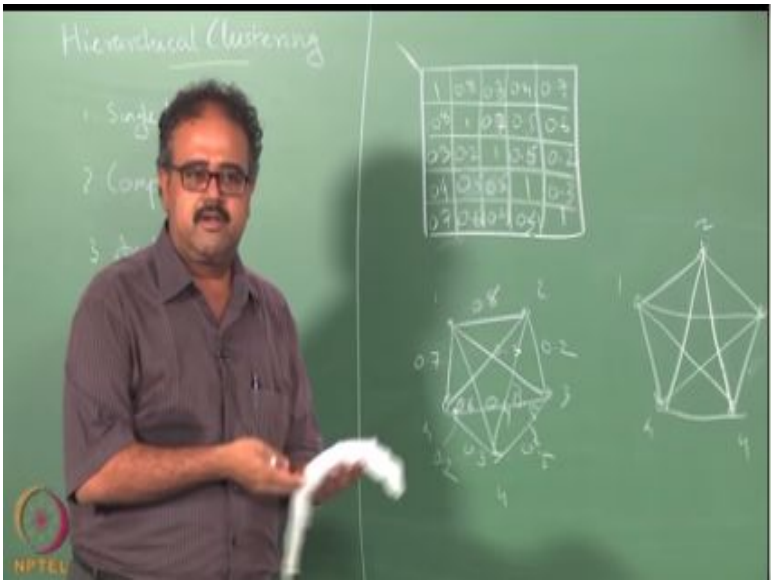
So two and four as well right so what happens now primarily everything is a cluster right, so everything has been I just stopped here, so this is my dendogram correct. So now i have done hierarchical clustering by using just simple graph theoretic concepts, what did I do? I just kept taking threshold graphs, and I look at connected components in that graph, and I got a hierarchical cluster right. So what is it equal to any one of those things I have written down there is it equivalent any one of those distance measures, I wrote down there.

Single link, how many of you think it is a single link? One how many of you think five thank you how many of you think it is complete link okay, one how many of you think is Average link? One and a half so I am taking the average, and somebody put their hand up like this somebody who did this so anyway. Single link right so the majority hope it was for single link 5 vs 1vs 1/2 right so single-link clustering right, so if we think about it I so what is the distance between the cluster 1 to 5and the cluster 3 4things, that are the closest right the pair of points that are closer so between four and two.

So that is mostly in the distance I am using right when I have the distances for and to this edge appears in the right and then it becomes connected right, that means that the closest points right these are the most similar so similar most similarly means closest distance is the smallest right.

So they appear in and therefore this is a single link clustering is equal not exactly equal into single-link clustering okay is it fine so can I erase that and I want to do this again except that I change the definition of cluster okay. Threshold graph will start off with this right, and I will take all the cliques in this graph as a cluster right, so what are the maximal cliques in this graph all of them right each one of them, so that is it that does not change so the same thing.

(Refer Slide Time: 22:02)



So I start off with a sort of with five clusters right then I do the threshold right, so the first page that appears will be this guy right now what are the maximal cliques in this graph one to write everything else is all by themselves, so again I get that I emerge this and the level is 0.8. Now I do the 0.7 level what do I get that okay so what are the maximal cliques in this graph is 12 and 15 but we are already inserted 12 as a cluster, so since we have not allowed overlapping clusters or anything here we are thinking of partitions here right.

So 5 will still be left alone right, so I have two possible cluster cliques here 12 or 15 right but since I do not consider overlapping clusters I cannot assign one to two clusters so I just leave it like that so at the 0.7 level I do not do anything, so point name we do not know anything, also I

do not do anything. Next, what do i go to 0.6 which is what 25 an okay, now what are we have maximal cliques here 125 right. So earlier at the 0.6 level nothing happened this cluster got formed at the 0.7 level itself here you have to go down 2.6 before the cluster gets formed okay.

Next what we do raise it 2.5 I do not need to put the right 0.5 this is what I get, so there is a new cliques that is formed like 34 right, this is what this is 0.5 right then you cliques that gets formed a 2.5 then what I do 0.4 that is a point for which is what one and 4 okay let us say change any rate no right I do not want to disturb any of the cliques that has already been formed unless a new clique is forming I do not want to break this and put it there or anything right so i will leave it like that. Then i go to a 0.3 what is the 0.3, 1 and 3 is it 1 & 34 & 5 what about three and five know what about two and three no right, so at 0.3 nothing happens but then I go to 0.2.

Now I finally have everything down right then basically 0.3,0.4 nothing happens at 0.2 I get the final merging okay it is fine, so there are two ways in which I can do is I can just think of connected components or I can so also think of cliques right and so what is the difference here if I choose to cut, remember I was telling you can cut that tree at some point and retrieve your clusters right. So now I can set myself a threshold okay I want to cut the tree such that the similarity between the data points in a cluster is at least 0.5 right, so then what do I do so I cut the tree just below that right like that and here I will cut it just here right oh well, in this case, it turns out to be the same sorry yeah.

So I said at least 0.5 right so 0.5 is a bad idea let us take 0.6 right I want at least 0.6, so what happens is right so I will cut it off here just below 0.6 I want at least 0.6 means I want it to be at least 0-600001 which I do not want it to exactly point it going to be slightly more than 0.6 I can't it just below 0.6 level here, so what do I get I get 12 as a cluster five as a cluster four as a cluster 3 as a cluster but if you do the same thing here just below 0.6 right I get 125 as a cluster four and three as a cluster.

So depending on how I did the clustering and how I built the dendogram given the same tolerance level my different clusters for me right so is this Plus this tie-up with any of the clustering technique that we already saw its complete link like why is it complete link? So I consider two clusters as having merged only if all points are connected that means, even the specifically the farthest most points also should get connected right. So the level at which I will merge the cluster now this will be the level at which the to the farthest point lie right so this is essentially this is the complete link.

So this is single link okay makes sense so you can always think of your data points lying on a graph and we can do all of these things but a nice thing about data points lying on a graph it can visualize them harm it is 2d that is interesting are all graphs visualize able in 2d, so they were named for it they call planar graph I mean you can still utilize other kinds of graph is just that they will have all kinds of course crossing lines right, so and so it becomes a little harder to visualize right but yeah huh this is planar complete not claim I don't think dendogram can visualize anyway come on doesn't have to be a graph right.

You can visualize the dendogram given any points it seems something even more important and when I start embedding them in some space and start giving you a distance measure it basically has to follow certain properties of that space right typically you end up wanting their distances to follow some kind of a metric right. When I start putting them in a graph right there basically arbitrary numbers I can fill in there right so I do not I can have some similarity measure which is not even a metric right, I do not have to worry about whether it makes sense whether triangle inequality is followed or anything again just take our can assign arbitrary.

Similarities to get the point okay and then I can say do clustering there might be applications where you need this kind of power, so that is the nice advantage of thinking about this as graphs right. So once you have these as graphs, then you can go ahead and do all kinds of your single

link complete length clustering or do minimum spanning tree do minutes whatever it is, and you can do your clustering. So that is the power of the graph modelling in fact so much so that when nowadays when I think of clustering applications I almost always think of okay what is the graph right that I can construct out of the data and once I construct the graph and then feed it into my clustering algorithm right.

So that is typically how many people operate right because there are so many powerful clustering algorithms there are based on graphs okay good any questions on this is not possible to reduce. So the complicity of the intelligence is it possible to reduce it most clustering algorithms are way more than order N2 in their operations because if you think about it right so if you use something like k-means and what is the problem with k-means? Every time the centroids change and I have to redo the computation to the centroids. Suppose I have n data points okay.

So every time the things change I have to do a for every iteration allowed to NK computation okay, and the number of iterations can be fab pretty large. so yes that is the problem right, but then k means in if you have effected a fairly large data set a small number of cluster centroids k-means is actually not a bad thing, for example, k medoids right think about the complexity of no it is humongous or Pam right, so when you do Pam basically you take each yeah exactly, so many clustering algorithms very expensive. So N2 is not too bad right but of course if you have a cheaper way of if you have a way of getting around computing the N 2 distances great.

If you have a better way of computer computing that is great. So what typically yeah I mean there are ways of doing that okay but it involves using very clever data structures and trying to reduce the amount of computation that you do right, by doing some cheap computations and then trying to do more expensive computations and so on so forth. We depend on the size of the depending on the volume of data that you are handling right and a number of resources that is available to you and so on so forth you might want to choose something over them it turns out that.

The overhead in doing this N2 computation is lot lesser then in some of those techniques is try to avoid the N2 computation right so one of the things which way should realize this big O notation is very deceptive right. Suppose you have ten elements in an array what is the best way to sort it likewise that might be instances where even though it looks N2 it might be cheaper to do that rather than try to set up something that is more clever okay, so that that is the thing you will not to think about but clustering is inherently an expensive operation there is no way around.

IIT Madras Production

Funded by
Department of Higher Education
Ministry of Human Resource Development
Government of India

www.nptel.ac.in

NPTEL

NPTEL ONLINE CERTIFICATION COURSE

Introduction to Machine Learning

Lecture-73
The BIRCH Algorithm

Prof. Balaraman Ravindran
Computer Science and Engineering
Indian Institute of Technology Madras

So the idea is I want to cluster large I want a cluster really large data sets right very large data sets and the way I am going to do it is the following right
so I am going to do a very rough clustering at the beginning right I am going to do a rough clustering at the beginning where I am going to produce tight clusters right, but many of them and i provide a lot of small, small clusters.

So essentially taking the suppose I have a million data points will reduce it to some 100,000 data points, but each one of them will be very, very small diameter clusters right I produce these things and then what do I take one representative point from each cluster, and then I try to do my hierarchical clustering on that
so initially I do something that little fast ends the fast and dirty right.

So I might get the correct clusters I might get a little bit off here and there but, but then I do a second pass through the data, and I will fix everything right so the way I do this as follows
I do one pass through the data produce some rough clusters right, and then I take the centroids of all these clusters right and then I do a proper clustering algorithm we only work with the centroids I do not worry about the million data points I reduced it to10,000 centroids 100,000 centroids.

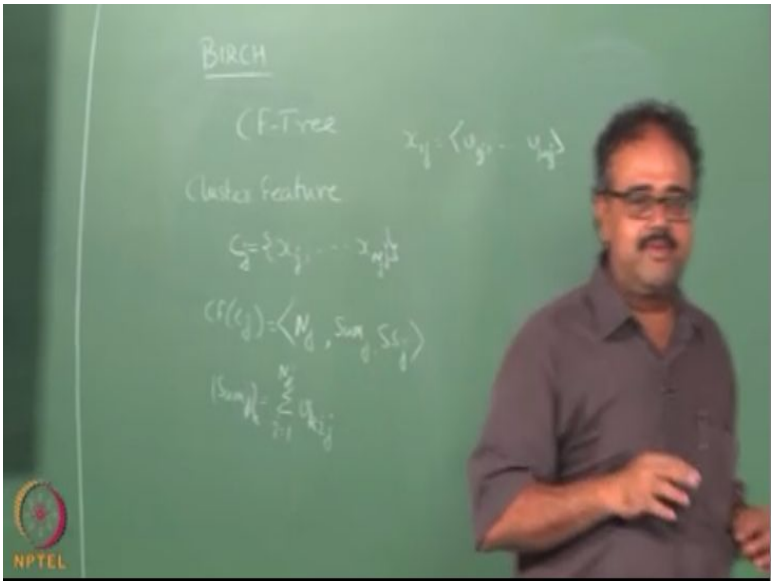
Whatever I only work with these 10,000 data points, so I know I know small enough that I can fit it in memory right memory is so large nowadays you can fit a million data points in memory but let us scale it appropriately right you have a billion data points and you are not able to fit all of that in-memory then reduce it to something small that you can fit in memory right so now what you do after you have done the clustering right.

You will have a new set of centroids for all your clusters right now what you do run through the data once more. I assign each data point to the closest centroid, so essentially you are making two passes through the data once to produce clusters which are very tight right and then you take the representative points and then you do whatever you are you can do an iterative clustering algorithm on top of that so once you are finished with that so what you do you go back and reassign the data points to the nearest centroid.

So the two passes through the data so that is one of the things that they should think about when you are working with very, very large data sets how do you minimize the number of passes through data right so if you can do it with one pass through the data great!
if you cannot we try to keep the number of passes minimal right if think of k-means rated very plain the k-means how many passes are you making through the data as many iterations.

As there are so every time you do an iteration, you need access to all the data points because I need to compute the distance of every data point to the centroid so I need to read the data all over again right so with a very, very large data just going through that might be plain if I can't store everything in memory right so if I want to read the data set all over again from the disk then it becomes a pain.

(Refer Slide Time: 03:26)



$$C_j = \{x_{1j}, x_{2j}, \cdots x_{nj}\}$$

$$X_{ij} =< v_{1j}, \cdots, v_{pj} >$$

$$CF(C_j) =< N_j, Sum_j, SS_j >$$

So we need to have something better than that right so what they came up with is a data structure called the CF tree came up with a data structure called these an of the tree where CF stands for clustering feature it is CF stands for clustering feature or cluster feature right so what is the cluster feature it is a three tuple.

Suppose I have a cluster right that consists of some endpoints like right so I will denote the cluster by seeing if the cluster C consists of some endpoints I will denote it as x1 to xn right in fact I should do something more than this I should say probably cluster j consists of cluster j consists of Nj points which I will denote as x1j to xnjj so it is consists of NJ point so this is the crystal so the clustering feature corresponding to the cluster J will be the number of points in the cluster okay.

And the sum of the individual coordinates of the data points suppose my x_{1j} is consists of some b_1 up till Vp right will be too many indices but so it is a point my x_{1J} is a point in a P dimensional space right so I essentially each one of them is a point in P dimensional space right so this will be essentially so my sigma ∑ is call the sum j.

$$(Sum_j)_k = \sum_{i=1}^{n_j} V_{hij}$$

And the I will come to that in a minute so some j sum j again a vector wait some j is a vector

and okay so some Kth coordinate of some j essentially the sum of the Kth coordinate of all the data points some of the coordinates in individual coordinates Sum j is a vector sum of the vectors

here that is my data points as vector concerns and sum of the vectors the, the reason I did not do that as sumof the vectors is sometimes people have been confused in the past.

When I read some of the vectors, they end up adding all the coordinates
I did not I do not want the summation ∑ also to run over k right so basically if you think of proper vector addition it is just the sum of all the data points right. Still, this notation is convenient for us when I want to write they will take the square of the individual coordinates and add them up, so these are this is what I call the clustering feature.

And the claim that they make in the BIRCH paper is that this clustering feature is sufficient for you to do some form of hierarchical clustering without actually not knowing the identity of the data points and suppose I give you the clustering features of two different clusters right is there any way you can compute the distance between the two clusters I give you the clustering feature of two different clusters.

I give you CFJ and CFI right can you compute the distance between cluster I and cluster J.
I've given you the sums the squared sums as well as the number of data points so if I divide the sum by the number of data points I get the centroid right I can go ahead and compute the distance between the centroids that gives me one way of measuring the distance between clusters that I do not need to know the individual data points all I need to know is the sum of the vectors that is one thing it said anything else I can do I am sorry I have sum square also we can use that I can do the radius right.

So I am going to leave that tears of homework okay show me that you can compute the radius in terms of some and SS right so I can actually compute the radius of the merged cluster also right based on the sum, the sum of the squared sums of the coordinates right so these are the things that I can do I can even do the diameter ok so I will leave you towards that out so you can do this you can do the centroid distance you can do the radius.

And you can do the diameter distances you can't do a single link complete link ok, so that is gone right but remember what you are doing at this stage is trying to the initial stage with the CF tree is trying to find small clusters the Custer's of small diameter so that is what you are interested but I am not interested in doing single link or complete link at this point I what I am trying to do when I use the CF tree.

The CF tree I will tell you how to build the CF tree a minute so what I am interested at this change is only in data reduction I mainly interested in making data reduction I am not really interested in actually finding the final clusters right so it is fine so we can use either the centroid measure or the diameter or the radius and then keep some kind of a threshold right so what do I do now is I start off from the root ok so the root is going to have so one cluster right.

So you going to have a clustering feature that is going to be there so as, as and when the data points come in the right I start updating the clustering feature at the root right the first data point comes in the clustering feature will be1 comma the data point comma the squares of the coordinates of the data point that will be the first one okay then the next, next data point comes in then what do I do 2, the sums of the two vectors right plus the sums of the squares of the coordinates.

But I do this only if the two points are close together and if they are too far away if the diameter is very far-right very large then I do not merge them I make it into a different cluster, and I actually do the clustering feature there right, so I keep doing this right I keep going until I come to some point where I have too many clusters that I have thrown it at this point right.

So essentially I start breaking the cluster into two breaking this leaf this node into two so some clusters will go here right, and some clusters will go here right and what do I do in

This place I insert a new clustering feature that summarizes all these data points enter a new clustering feature that summarizes all these data points so why am I doing this so when a new data point comes in the right.

I look at these two clusters and figure out which of those two clusters is closer to this data point ok and then root it down the trees right so if I keep all the clusters at one level then every time a data point comes in I will be doing a very large linear scan to find out which cluster the data point should belong to does it make sense and suppose I suppose I start off with one cluster then I produce ten clusters.

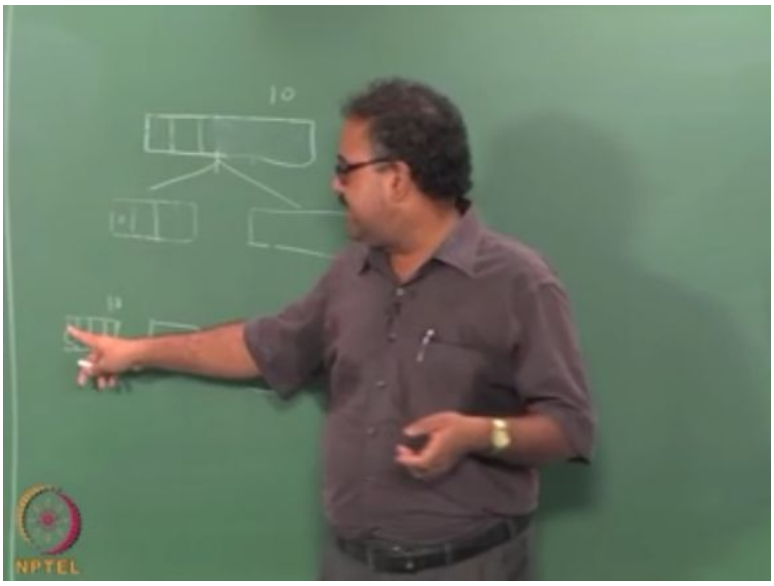
Now a new data point comes in I have to check against all the ten clusters to see which is the best cluster to put the data pointed right now what I am trying to do is organize this in a more efficient manner so that instead of taking the order of 10 right it is going to take some logarithmic factor here when I have to make two comparisons will take order of two now depending on how many I put in here.

So I suppose I divided into two and then put five here and we will take order of six right, but then it is a tree becomes deeper then you can see the savings right the savings will be tremendous, so it is you have some kind of a logarithmic compression, so that is essentially what we do here yeah okay so initially what I do is I have 1 node okay.

So I start keeping track of all my clusters here right suppose I reached in clusters let us say 10 is the maximum it I reach 10 clusters a new data point comes in I want to create 11th cluster sorry various factors it depends on what is your memory size right so and in fact, you have to go even further down actually so it depends on your page size not just on in-memory size right want your every leaf to span multiple pages right because a single axis could actually lead to more page faults right.

So you want to see if axis one entry in a leaf I would really like the entire leave to come into the and stay there so there are all these kinds of consideration when you start talking about really large data right you have to start going into the system level of things right so you have to know how things are stored you have to know how things are accessed right, right so here let us say I pick 10 for somehow right I said 10 is the number I am going to store and what are these numbers.

(Refer Slide Time: 16:05)



These are essentially three things like this right each one of these only three things like this at some point I say okay even if have a 4kb page even like, like 100 clusters I should run out of space right I will say no, no I want to break it right so then what I do is I split this and I put 10 here and I say another 10 here I am sorry put a 5 here put another 5 here right and then I will start off with two entries here right instead of having 10 entries it all go down I will get two entries here right.

And each one of them will be so the first entry will be summarizing the lab the first child the second entry will be summarizing the second child right now a new data points come in I can try

to shove them into this right and then at some point it will come to a point where this thing gets filled up then I will split it into two right so I can do one of two things I can actually push it further down right or you can split it into two and add another entry.

There which is better not by this like 10 rate I mean I have 10 I have space to store 10 CF features there right so I could either make it broad branching at all I could make it deep which is better broad it better be deep is better draw this better so if you choose to go broad when do you go deep then what gets filled up when this gets filled up and then you only push it down from there is it, no, no yes right that is ready to do its work done.

But do not doubt yourself what you said was correct okay I just wanted to see how sure you are up to your answer so essentially you keep broadening it right, and this gets filled up then again you split that everything goes down one level okay so keep doing this and until you have gone through all your data points.
So that the small point here is to notice, I am not doing a numeric example and if people are interested in a numeric example of how BIRCH works.

You can refer to the BIRCH paper so we will put up the BIRCH paper online right we will write? Yes! We will put up the best paper online, okay. So what we what will happen is suppose a data point comes very, very beginning right, so it will get associated let us say with this cluster so, and for that cluster, I compute the clustering feature added to the clustering feature in that cluster right but then later on as I keep going down more and more data points will come here.

And some more data points will come here and so on so forth it is likely that if you if, I try to insert the data point again into this class three right it may go into some other cluster because things move so there is this significant order effect right so sometimes what people do with the BIRCH is after they grow on the tree to some point they stop okay, and then they try to do some kind of rebalancing okay that is just an audition optimization step.

You need not do the rebalancing step you could grow the CF tree in one shot right more often than not it will work so the end of the clustering what you will have is you will have a lot of right lot of leaves at the bottom and each one of them containing say ten clustering features right and these are the final clusters that you want reservation or these your final clusters know right.

You remember I told you the first phase when I am constructing the clustering features the sea of the tree all I am interested in finding on tight clusters that I can then use as representatives for doing a further clustering right so what do I do now is go into each of these clustering features compute the centroid I can do that right divide some by Nj so corresponding to each entry in each one of the leaves can get one data point which is the centroid right.

And then I start with these centroids and do a clustering all over again here I can do whatever I want I can do single link I can do a complete link whatever so because I reduce the number of data points is something small and manageable right I can do whatever clustering method I want on these data points right and then what I do is the final centroids I get after I do that clustering I go back and look at the data all over again right.

So I can the data once when I build the CF three right now is can the data again after I finish my clustering okay and then assign the data to the nearest centroids you need a new point in you can just as any recording video now I need to cluster the whole data right I need to know what cluster you belong to the right so I need to know that right so like I said I cannot trust the identity here.

So initially I said okay this clustering feature got the data point I now so the same clustering feature would have percolated down to the leaf but I cannot say that okay this clustering feature.

whatever cluster it goes to that is the cluster that the data point belongs to because that would have drifted away this is the centroid could have drifted due to the later data points.

So I cannot do that is not the, the mapping between the CF the clustering feature and the data point is not, not static it would have changed or drifted away. Therefore I have to do the second round of assignment people understand BIRCH say I really like BIRCH you know because it is a very simple algorithm and it also allows you to do handle very large volumes of data I usually ask people the implement BIRCH in one of the programming assignments okay

IIT Madras Production

Funded by
Department of Higher Education
Ministry of Human Resource Development
Government of India

www.nptel.ac.in

NPTEL

NPTEL ONLINE CERTIFICATION COURSE

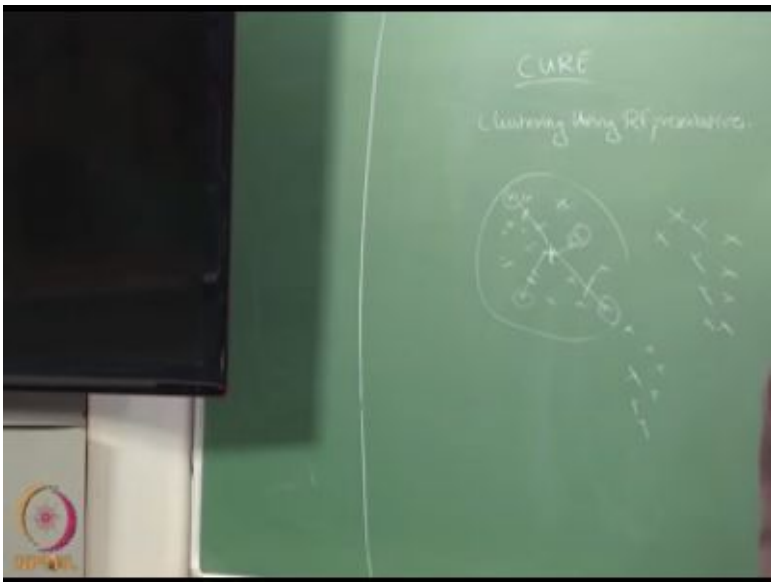
Introduction to Machine Learning

Lecture-74
The CURE Algorithm

Prof. Balaraman Ravindran
Computer Science and Engineering
Indian Institute of Technology Madras

Cure sorry it is its abbreviation yeah the clustering using representative points

(Refer Slide Time: 00:57)



Okay fine so with CURE again was touted as a an algorithm for handling lately large datasets that CURE again this is an algorithm for handling large data sets, so the way you do this here unlike BIRCH where you actually go through all the data points so CURE what you do is you sample a large fraction of the data okay, as large a fraction as would fit in your memory

comfortably sample a large fraction of the data and then you do some kind of clustering on that right.

So what you do in the clustering then you after you've done your clustering right, so you start off with some initial clustering of the data right, what you do is let us say I have something like this right some kind of weirdly shaped data like this side once I have done some clustering so I will take the centre okay, I will take the data point farthest away from the centre-right so this is let us say this is a cluster I take the data to point farthest away from the centre let us say that is it okay next I take the data point that is farthest away from this data point.

In that then I take a data point that is farthest away from roughly do not worry I do not measure it with a scale or anything right I will take the data point that is far the farthest away from both of them to put together right, so I compute the distance from both and then add it up and then, let us say that one and then I figure out one more if I need it one more right I will take that one right so now I will take these as may represent that one and take these as my representative points for the cluster.

Right, so a data point gets assigned to that cluster which has the closest representative point basically I am looking at the boundary points, right I am looking at the representatives or things that delineate the boundary of my cluster right I do not take too many points right I take some number of Representatives lately like 3 or 4 representative points at all or say ten representative points so that they do not completely trace out the boundary, but they give me some idea of what the boundaries okay.

So this is why representatives I reassign the clusters I reassign the data points right so I have some clusters, now I find the representative points I find the representative points for this cluster I find it for this cluster now I reassign the data points to clusters so a point goes to the cluster

which valve whose representative point is closest earlier we used to reassign them to the close a Centroid now will be assigned them to the closest representative point, now for each cluster I will have 4 points right for each cluster I will have 4 points.

So what I will do now is I will forget the clustered memberships right this is just like how k means works right so k mean what will I do I will reassign a data point to the closest centroid now I will reassign the data point to the cluster that has the closest representative point initiate whatever the whole thing is done out of a set of samples, have a set of samples I drew from the data for points for each cluster Oh yes, among the sample post but this one thing I forgot about the representative points1 point I forgot about the representative points apologize.

So once you identify these guys right so you do not take them as your representative points because they become too susceptible to outliers okay, you do what is called shrinking right you move them some fraction alpha α towards the centroids, let us I find the points at the boundaries right, and then I shrink them a little bit towards the centre-right so that I do not get too influenced by the outliers, and the shrinking is done by a fraction of the distance of the centroid so that the farther away for you are from the centroid.

So the greater the shrinkage, okay, so this hope you find the representative points, and once you find the representative points, you go ahead and keep doing this on the same sample right. So the idea of taking the sample is the sample is small enough to fit in your memory right so that I do not have to go back to the disc to read the data for the second iteration they still a couple of stages to go this is the first stage of CURE at all sample I cluster using this representative points okay, so remember that the sample.

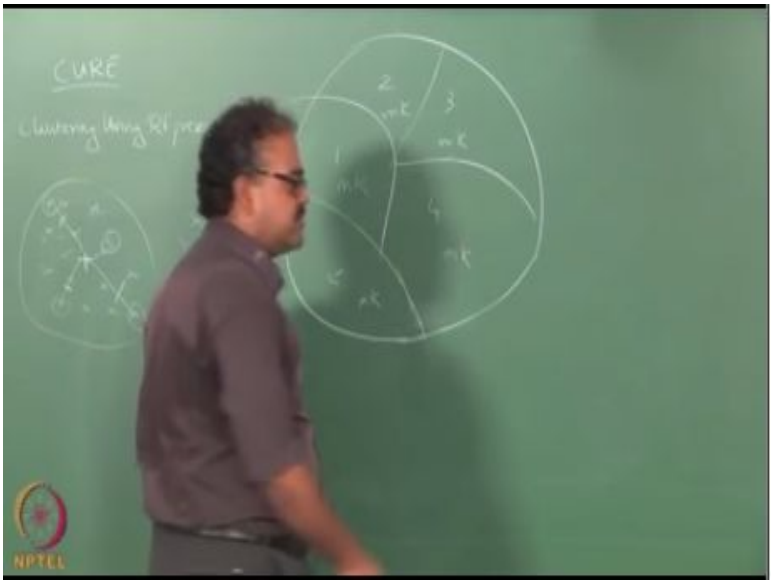
I hope there is representative of the whole data set, but it might not be right I have taken some as large a sample as if can from the data right into my memory, and then I do this clustering around

representative points right is it the clustering the representation based clustering is clear right, just like k-means instead of assigning it to the data point with the closest centroid I send it to the data point with the closest representative. It is a parameter yeah cm some m you choose like, in this case, we chose it to be four right, yeah was it as I said you start off with an arbitrary clustering you start off with some arbitrary clustering okay, and then you start pick representatives find the centroid of that clusters that you have right and then go to the corners shrink them right do not forget the shrinking step.

Shrink them and then you get a representative right, so just like the centroids are not real data points the representatives also need not be real data points okay because once you shrink them, they are no longer so when you find them they are real data points then you shrink them then they are gone right they are no longer real data points okay, now you get m such sorry I will be using all the data points of the point but only the sample, so we did not bring the clustering I am not worried about anything else.

That I have not sampled right now what I do is I keep doing this until I have converged to some clustering right once you have converged to some clustering, okay what I will do is I will remember those representative points suppose I have k clusters I will have M times k points it will remember those k representative points I will forget everything else right, now I will pick another sample from the data I will repeat this, in fact, the recommended way to do this is to partition the data initially randomly into some large number of bins like say k bins at partition the data randomly into k bins and then do run CURE in parallel on each one of those K bins.

(Refer Slide Time: 08:47)



So if you have a lot of machines so what you can do is you can take your entire data set, okay this is not meant to be a geometric representation of the data right, so now when I say I am going to divide it into two parts, I am not saying take data that belong to one part of the input space and assign it is I do not know this right this is randomly split the data right and say that okay here there are five bins again in each bin I will run CURE independently right and what we will end up with some mk points for each of these bins.

Now, what will I do is I will throw in all the mk points into a single clustering problem and one CURE on that mk what about 5, 5 times mk points again, again you will get a set of representative points right for each cluster will get a set of representative points then I will go back and assign each data point to the cluster which has the closest representative point right and on each part I run CURE right I will end up with some clusters right and for each cluster I will have a set of m representative points.

Let us say I end up with k clusters in all of them I will apriori define k right I will end up with k clusters in all of them right and so there will be some mk points from here where m is a

parameter that I have chosen already which is the number of representative points let us say 4 now what is happening is I have a small fraction right for each cluster I am going to have 4 points is a small number with my number of data points will be very large the number of clusters will be small right I will have 4 times.

The number of clusters a small number right I have now 5 such divisions so I will have 5 times m times k points case number of justice k clusters just like k means you define k, so but the point here is at every point at no point am I looking at a larger clustering problem right so I will split the data right so I am looking at some fraction of one-fifth of the data or one-tenth of the data, and that is the largest data set size I am going to look at this is why CURE is an algorithm for handling very large data.

So I can take a very, very large data I can split it up 10 ways the second spirit of 15 ways 20 ways and I will do each one of these clusterings. Hence, if have if I am looking for 100 clusters and for representative points, so that is basically 400 points returned from each one of these clustering's right and then what I do is 5 of these right, so I basically end up with 2000 points right, so I have100 clusters for representative points per cluster right and 5 such problems I have solved so end up with 2000 points.

Very, very small number right so I go and run CUREon that again right again I will get 100 clusters on this 2000 points right, bad idea 100 classes on 2000 points is a bad idea end up with 20 points per cluster, so what you should typically be doing is whatever final clustering you want to end up with this should be much larger than that, this K that you use here suppose I want to end up with 30 clusters then I can use 100 clusters suppose I want to end up with 100 clusters I better end up using a larger number 200 clusters or something okay.

So that I love enough data points to cluster finally and then I do the clustering suppose I end up with 30 clusters each will have four representative points I will pick those four representative points I will do another scan through the entire dataset right how many scans I have done through the entire dataset so far one I had done only one when did I do the scan? To do the partitioning right, so I did the scan once to do the partitioning the second time I do the scan I will have m representative points from 30 clusters.

And I will go back go and assign the data point to the cluster that is the closest representative point right, so at every point, I do not solve the problem that is larger than what I put into this one partition this allows me to do these things very rapidly. If I have multiple processors that can run right, I can do this in parallel this stage can be done in parallel and whatever cluster representatives it returned and then I do in a second round. Hence, CURE is if you do the implementation correctly.

It is rather fast another nice thing about CURE is that because I am doing this clustering around representative points I am not really limited to looking at convex clusters like I have with k-means rights okay means if I have two centroids right so basically this will be the separating FF three or four centres I will basically be looking at some convex shape around the centroid right, but if we have since you have multiple representative points, I can actually have non-convex shapes also disadvantage of using non-convex shapes this is overhead right so if the data is small you really do not want to get into the CURE kind of setup right because the overhead is large.

And you have to maintain so many representative points right so every time finding a representative points involves you getting the centroid and doing this computation to go to the edges and then shrinking them right so it is additional overhead that k-means you just find the centroids and you just move on here you have to do an additional computation, so that is the overhead okay good, so any other any questions on this I said a question is no it usually defeats

the purpose I am assigning it to the data point that has the closest representative point right so the centroid is will be anyway surrounded.

By the representative points, I am taking so it will anyway end up going to the same there might be a small change here and there, but people typically do not include the represent the centroid, in the representation, this is that we have now to find a set of two parameters of pair? Yes but you get some advantages was it right it runs on large data sets which is stop and think about it you could do and claim it run some clusters right, so the and the second thing is you can get non-convex clusters.

So if I have fun funky shapes, funny shape clusters, then you get those in a sense? Representative points are not real data points they are fictional points like centroids so when it does the reassignment of points to clusters, and representative points never get reassign I mean they are not points at all remember I told you to go to the edge you find real points, and then you shrink it by a factor α towards the centre, so just another parameter we have to select so three parameters now it is drink it by some factor towards the centre so, therefore, they are not real points just like the centroid does not really get reassigned neither make the representative points.

So the whole point of doing the shrinkage was the reverse the outliers right, now you could think of using the mid idea more computation without replacement picture yeah without a replacement that is why I showed the partitioning typically sample without replacement and CURE you could I am not saying you cannot I mean see the point is for every, every variant that you propose you have to either convince me empirically that your variant is better or convince me theoretically that your variant is better.

No that is this let is go feat will generate papers right and let me stop and think what is it that it buys you know is there some kind of statistical parameter that will become better by sampling with replacement is a sampling without replacement for example will you get more stable cluster

estimates if you sample with replacement, so regardless of what how many ever how do I speak the samples right so if you say that I will get the same cluster centroids maybe then that is a valid thing to do right if you cannot show anything like that.

Then not clear right, so the thing with which has more overhead sampling with replacement or sampling without replacement has more overhead really? With replacement has less overhead if you are only sampling, okay if I am interested in partitioning the data I can just look at some random permutation of indices right and then just chop the data at some points right so that is like sampling with sampling without replacement, but I have exhausted the entire sample space.

So if I am going to do that for sure right then it can do it efficiently right then that will actually have lesser overhead than sampling with a replacement but why the sampling with replacement have lesser overhead than without replacement when I do it with replaced with the replacement I do not have to keep track of what I have already sampled from doing without replacement I actually have to remove it or allowed to have some bit that sits there.

Let us say this string has been visited so how now how do I change my sampling distribution so that I avoid those data points which I have already sampled yeah, what is it now you can you can pre-roll your sampling yeah so if you are going to just sample one at a time then it is a then it is a problem right if you can pre roll your sampling that is essentially what both of you are saying that you can what I am saying is you a priori generate all the random seeds you want right just to the permutation and then just keep chopping off the whatever is the ID at the topmost is a way of implementing that can be pretty efficient.

IIT Madras Production

Funded by

Department of Higher Education

Ministry of Human Resource Development

Government of India

www.nptel.ac.in

NPTEL

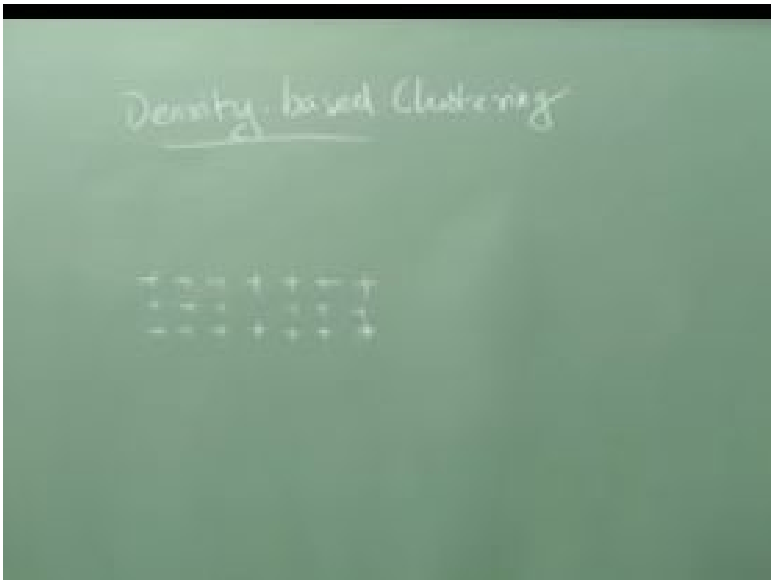
NPTEL ONLINE CERTIFICATION COURSE

Introduction to Machine Learning

Lecture-75
Density Based Clustering

Prof: Balaraman Ravindran
Computer Science and Engineering
Indian Institute of Technology Madras

(Refer Slide Time: 00:17)



What are the two classes as a row of pluses there in the row of pluses at the bottom, so there are two clusters here right.

So we will not touch these data points right

we do the following,

now do you see two clusters side by side so what essentially defines clusters is not the distance between the data points that are more like the density of the data points.

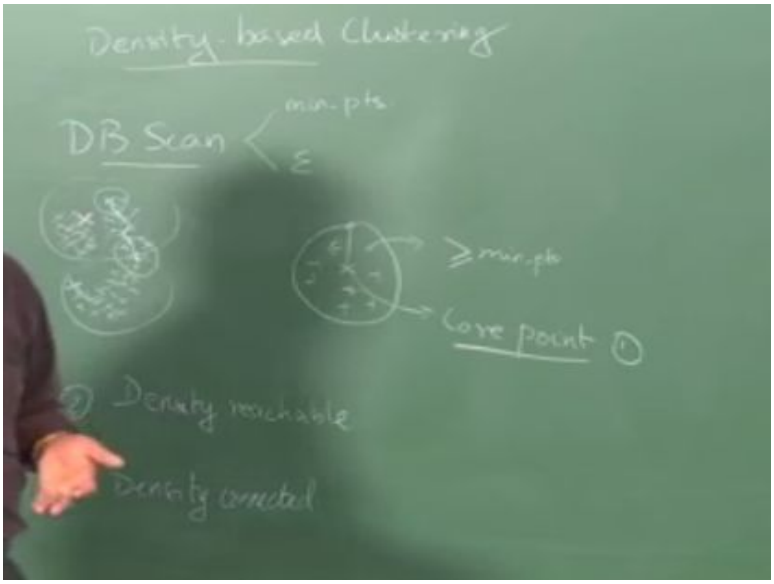
Naturally, when we think of clusters, it is really that where data points are dense is one cluster. Then we tend to draw the boundary between clusters where the density is lower right so we did not change the data points in the initial set of data points you are very happy to say that okay the cluster or the top and at the bottom right so what cost you to change your clustering when added the additional data points.

The density went upright the density went up differently, therefore, you said oh okay the least dense point is no longer between the two but vertically right no longer horizontal but vertical was the low-density point right.

If you run k-means, you might get anything we do not know and depend on where you start with right.

So the question is if you have your intuitive notion of clustering has to do with density so why do not you try to come up with a clustering algorithm that captures this notion of density right and try to come up with a clustering algorithm that captures this notion of density.

(Refer Slide Time: 02:45)



So there is a very popular clustering algorithm called DB scan,
It's a very popular algorithm called DB scan that does density-based clustering okay so Dbscan has a lot of terminologies that they define right once all the terminologies are described then the clustering algorithm itself becomes nearly trivial okay.
but the terminology takes a while to get through, so the basic idea is very simple right
suppose we have data points like this right.

You see two clusters here clearly two clusters right incredibly hard to get k-means return these two classes and if you run k-means what will happen is you will end up with the centroid somewhere here right another centroid somewhere here right and it will say these data points are one cluster and these data points another cluster is the biggest drawback with k-means right.
so what DB scan says is that two points belong to the same cluster okay.

If we can get from one point to another right by moving only through only through dense regions only through points that are close by right two points belong to the same cluster right suppose we take let us take let us say we take this point let me take this point right in fact they look pretty far away and if you look at the direct distance between them they are pretty far away in fact the

points of other cluster which are closer to this than this point right but when you look at it you think that they this thing is one cluster this thing is another cluster.

So what is the intuition here is that can keep hopping at no point do we take a very big hop right we can keep hopping to things that are nearby and we can go from here to here right, so if you take these two points the blue and the brown one right so if we take these two points the no way we can hop from here to here right because there is a this nice gap here right there is no way we can get from here to here only by going through dense regions right is that clear.

So that is the intuition that we are trying to capture here so what is it that we should define now first what we mean by a dense region right so that is essentially what we have to define so we will start off by defining something called there are two parameters that DB scan uses one is called min points other one is ε (epsilon) there are two parameters so min points essentially gives you some kind of a threshold on how many points would you consider as being dense right.

And ε (epsilon) gives you the area over which you will perform the count is a min point says okay if you have five points okay then you are in a dense neighborhood but where do we count these five points okay in a radius of ε(epsilon) around me you count the five points okay count in the area ε (epsilon) around me if you find five points then you are in a dense region and if you make the ε (epsilon) very large then it might encompass my entire input space then everybody will be dense so it does not make sense.

The ε (epsilon) has to be small likewise if we make my min points 1 point 1 not 2 points that means everything will look dense unless make me ε (epsilon) very small so these are actually complimentary things we can control my min points and we can make them in points very small right and then make the density high right or we can make my min points large with the larger ε (epsilon) and that it then also we can make me density high the effects that you will see are different for both right so we let you think about it.

Which one we mean what is the effect of increasing min points versus decreasing ε (epsilon) okay so essentially what we am saying is okay take a data point right take a radius ε (epsilon) around it okay that that is that is a circle so take a radius ε (epsilon) around it and count the number of points okay if this count is greater than min points okay then you call this a core point take a data point take a ready ball of radius ε (epsilon) around the data point right.

Count the number of points right if the number of points is greater than or equal to min points right number of points is greater than or equal to min points in that ball of ε (epsilon) then you call this a core point right a core point is a point that lies in a high density region that is the definition we have right.
So we say a point is you see that the point is density reachable okay.

A point is density reachable if there is a core point from which you can reach this point it went by traversing only through core points so this might not be a core point because is it the border rate we draw in the radius around it we get only one point here okay so this might not be a core point right but then if we start here let us say let us say this is a core point for sure a lot of points in the anything from here i can basically move two points within ε (epsilon) of itself.

Which are internal core points right so we can move to you can move to core points right and finally reach this that no point we should be making a jump greater than ε (epsilon) right because it is starting a core point we will have enough points in the neighborhood that we can actually jump to something within ε (epsilon) right so I make steps of size ε (epsilon) and I actually go through core points every time then we call them density reachable say point I is density reachable if there exists a core point from which I can reach here.

By jumping only from a core point to core point until the last step obviously every core point is density reachable because it Is reachable from itself right every core point is density reachable and then there will be these border points right which are density reachable from core points

there might be other points which are not density reachable from core points which are essentially outliers okay right.

So these are the definitions we have so this is the first right these are the two quantities we need this is these are not definitions this is the first definition what is a core point second definition third definition is density connected case I will say two points I and J are density connected if there exists a core point k from which both of them or density reachable.
That makes sense right so what is density reachable. I start from a core point and only move to core points right until I make the last hop to this case so no point I will be making a move greater than ε(epsilon).

And all the points I visit in the on the way will be core points ok that is density reachable density connected is if I and J that exist one core point from which both I and J are density reachable then the I and J are density connected okay
here is the next thing. So I and J are in the same cluster if and only if they are density connected. this is the definition of a cluster two points I and J belong to the same cluster if and only if they are density connected make sense?

Sorry! How do I implement it ? this so I start off with YS any point right I pick up a random point right I figure out whether t is a core point or not right then okay it was a core point great so I will keep that as my starting point for the cluster how do I determines the core point I pick up a point look in the neighborhood figure out if there are ε (epsilon) if there are min points within a neighborhood of Epsilon if that is the case then I will keep it right.

Then what I do is I look at all the neighbors of that point and look at all the neighbors of the point and each point in turn I will check whether it is a core point or not right, so each point in turn I will check if it is a core point or not so any additional points I encounter when I do this check I throw it into my Queue so I will keep going right if I reach a point which is not a core point okay so I will not insert the neighbors of that new neighbors of that point I will just stop

there if we reach a point is not a core point I will just leave that exploration go back to my Queue to see if anything else is still there.

So I keep doing this until my Queue becomes empty so all the points I have examined from the time I started till my Queue became empty go into a single cluster sorry like it that first search moves like a depth first search you do that all these data points go to a single cluster now what I do I go and start at a random point which has not been assigned a cluster so far and then do it if first search again till I find the whole clusters I do this and I am done.

So the nice thing about this is I am really doing only one scan through the data right so every data point I will actually look at it once right I will examine the neighborhood right and then I will go on but then the number of computation I will do will be still significant so DBscan is a slow algorithm even though I examine each data point only ones but the amount of computation I do an examiner data point is significant.

Because I am looking at the radius ε (epsilon) and then we have to find all the neighbors within the radius so unless I have a very efficient data structure that will return to me the nearest neighbors very quickly right so this can take a significant amount of time in running so there are some efficient implementations of DB Scan out there it is really cool in that it gives you all kinds of arbitrary clusters right.

And so these kinds of things today whatever I drew that that you would not be able to recover using k-means or even hierarchical clustering depending on the kind of cluster measures that you choose right cure might be able to give you or might not be able to give you this kind of clusters depends on again how your sampling that you do and what you start off with and so on so far so whole bunch of imponderables.

But again the same thing with DB scan depending on which is of min points in ε (epsilon) right you might get very weird results, right yeah so if you look at the data mining text book by Han

and camber right Michelin Cameron book right so they have actually horrendous examples of DB scan in fact I think they did a significant optimization to find out which are the worst parameters possible form in points in ε (epsilon) and they give you results.

Because they wanted to write a paper that said hey we have an algorithm that does better than DB scan so they said oh DB scan can perform really badly if you give it bad parameters so let us give it back parameters and then we will beat it right. So I think this is assuming that they were actually being more fair than that but the way they look at the results it looks like that I mean DBscan looks really horrendous and they are a method which is called chameleon.

It is an acronym yeah so I think C stands for clustering I am not sure good I sure about that either so yeah so yeah it is a nickname DB scan right I just start off with some arbitrary point and then I look at look at the exam examine the clusters and soon so forth, so what happens if I happen to start off at a boundary point instead of a core point I will examine it I will say okay it is not a code point I will throw it away right so it will never get assigned to any cluster.

So it will kind of be all by itself as an outlier even though it should belong to a particular cluster that is one thing the second thing is suppose I want to vary ε(epsilon) and I want to run the clustering again I basically have to do it all over from the beginning right so what optics does this gives you a clever way of ordering your data points such that right for different values of ε (epsilon) you can recover the clusters very quickly for the same value of min points min points is fixed but ε (epsilon) changes right.

So it gives you a way of ordering your scan through the data such that for different values of ε(epsilon) I can recover the clusters very quickly and is a really cool idea right.

IIT Madras Production

Funded by

Department of Higher Education

Ministry of Human Resource Development

Government of India

www.nptel.ac.in

NPTEL

NPTEL ONLINE CERTIFICATION COURSE

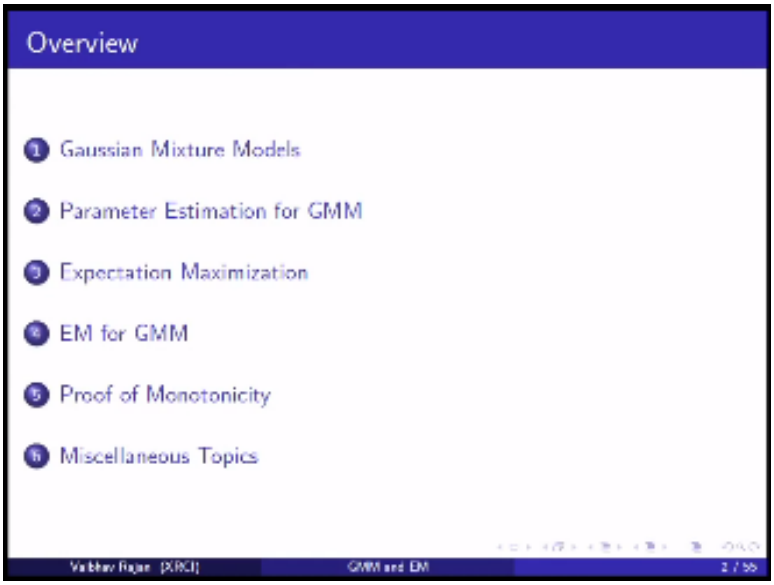
Introduction to Machine Learning

Lecture-76
Gaussian Mixture Models

Prof. Balaraman Ravindran
Computer Science and Engineering
Indian Institute of Technology Madras

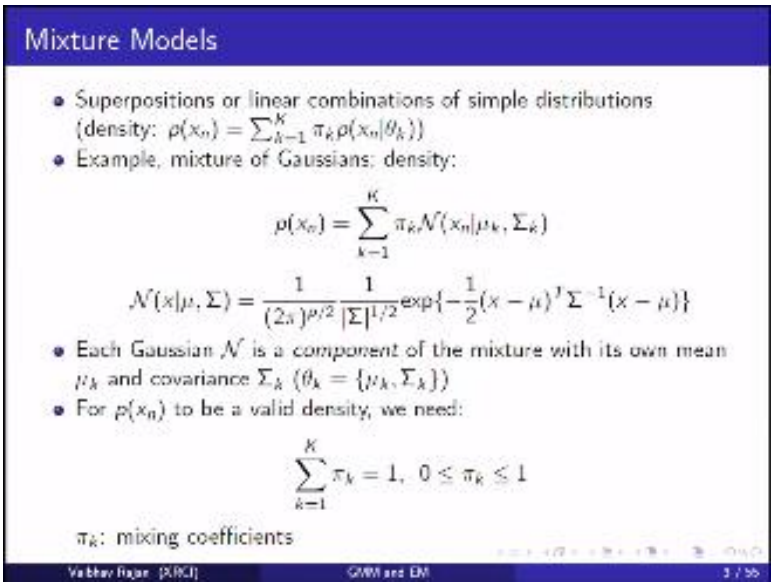
I will be talking about Gaussian mixture models and the Expectation-Maximization algorithm.

(Refer Slide Time: 00:21)



The plan is to start with introducing Gaussian mixture models and then talk about mainly how we estimate parameters for a Gaussian mixture model and then through that introduce what Expectation-Maximization is because that's the iterative algorithmic framework that we will be using for parameter estimation. That's in general and then we will come back to Gaussian mixture models and see how EM can be used for Gaussian mixture models and then talk a little bit about theoretical properties of EM and why it's interesting.

(Refer Slide Time: 01:03)

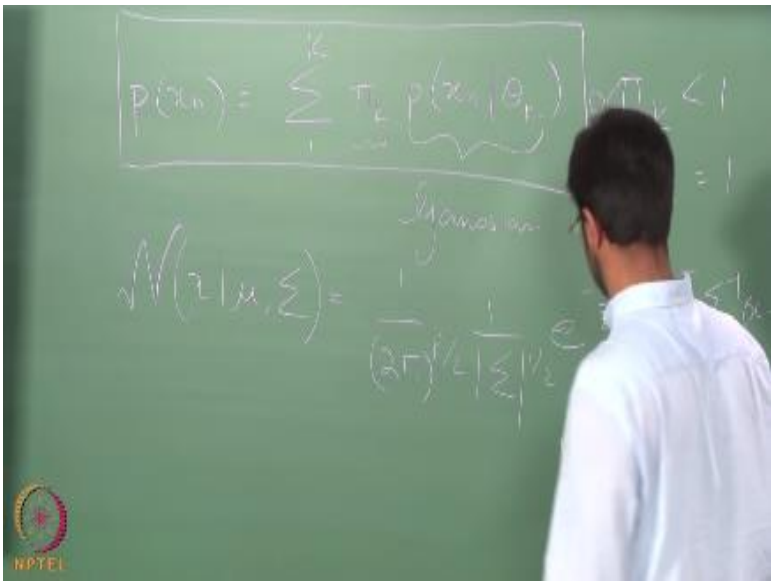


Mixture models, as the name suggests, are a mixture of models. Formally, they are linear combinations of distributions. So they typically have a form like this:

$$density: p(x_n) = \sum_{k=1}^{K} \pi_k \, p(x_n|\theta_k)$$

The density of a mixture model is a linear combination of other densities p and different mixture models will have different forms for the probability distribution here.

(Refer Slide Time: 01:50)



$$p(x_n) = \sum_{k=1}^{K} \pi_k \, p(x_n|\theta_k) \qquad (1)$$

The density is given by a linear combination of different probability densities and here we have k components and each of these components have a mixture weight (or a mixing coefficient) denoted by π_k. So this probability here can assume different parametric forms. The most common is the Gaussian. And when it follows a Gaussian, this is called a Gaussian mixture model.

The Gaussian mixture model is one of the most commonly used mixture model. It can seen in a lot of different domains such as bioinformatics and speech processing. One of the reasons why it is used is because it is mathematically tractable but there are other nice properties too. I guess you all know what a Gaussian is but let me write it down anyway because we will use this very often in the coming few slides.

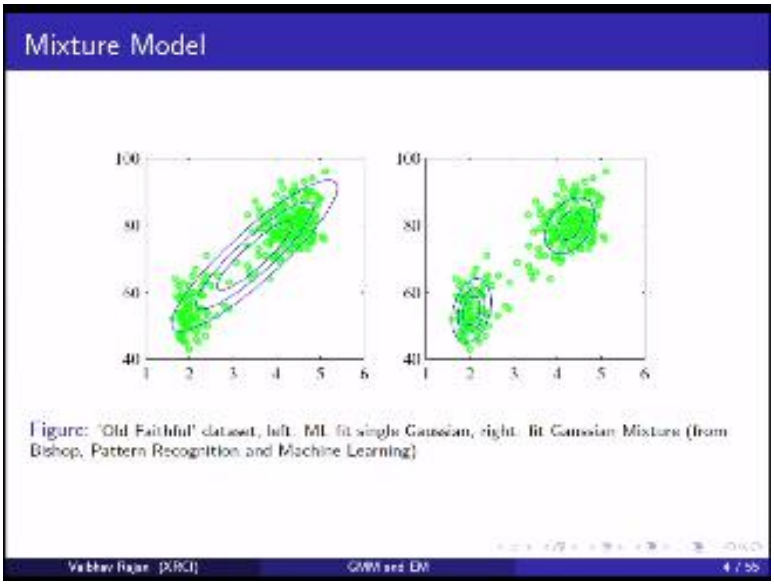
$$N(x \mid \mu, \Sigma) = \frac{1}{(2\pi)^{\frac{p}{2}}} \frac{1}{|\Sigma|^{\frac{1}{2}}} \, e^{-\frac{1}{2}(x-\mu)^T \, \Sigma^{-1} \, (x-\mu)}$$

So this is the form of a Gaussian. I am assuming you all know this but let's keep it.

So each component here is a Gaussian and each of these Gaussians has its own parameters - the mean parameter and the covariance parameter. And for equation (1) to be a valid density, we need the π_ks to be between 0 and 1 and also the sum of all the π_ks to be exactly equal to 1. We can show this mathematically:

$$\sum_{k=1}^{K} \pi_k = 1 \quad , \quad 0 \leq \pi_k \leq 1$$

(Refer Slide Time: 04:20)

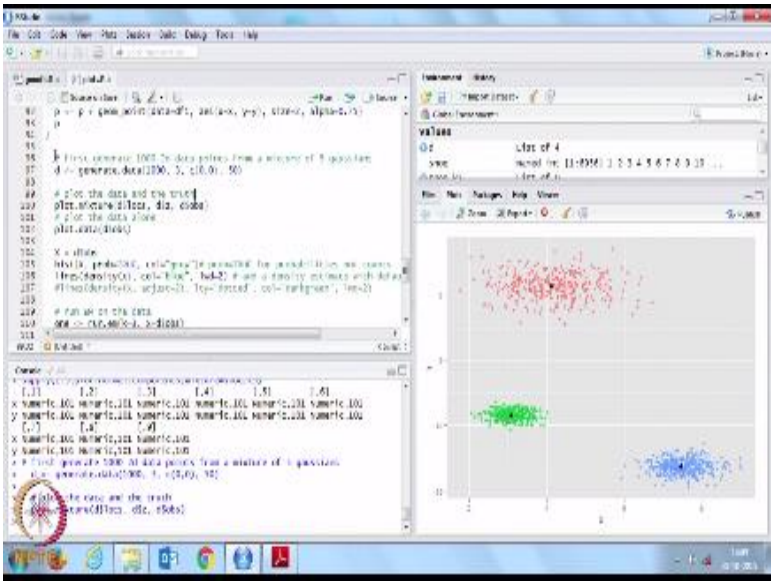


So why do we need these mixtures? Why do we need these superpositions of densities? Here is an example from the Bishop's book. It's a very well known data set – "Old Faithful" dataset. It is a two dimensional dataset and we have plotted the dataset in green points. On the left, we try to fit a Gaussian on the dataset. Now, visually it clearly looks not okay because the Gaussian is most dense around the mean but when you see the plot on the left it doesn't look like the data is most dense around the mean.

But instead of a single Gaussian, if we use two different Gaussians and try to fit a mixture of two Gaussians to this data as shown in the plot on the right it looks somewhat okay. The data is dense

around the means of both these Gaussians and it looks like a mixture of two Gaussians would be a good fit for this data. So let me show you some more examples.

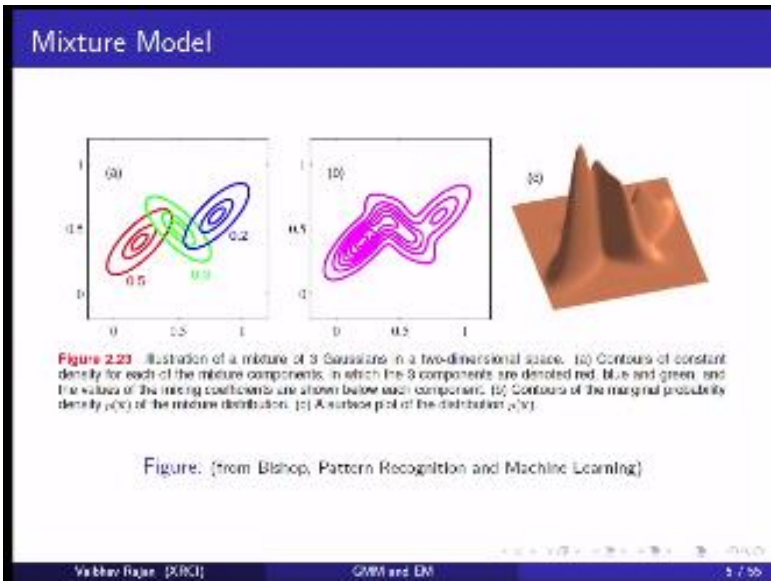
(Refer Slide Time: 05:49)



This is some R code to sample data from a Gaussian mixture. I am going to sample and then plot the data. Since I have set the number of components to 3 in this case, every time I sample data, it is sampling from three different Gaussians and you see a clustered kind of data which has three clusters.

If I change the number of components to 5 or 6, you will start seeing more clusters. The clusters need not be well seperated as shown here but can be overlapping as well. The first thing that comes to mind when you try to model data, with such a clustered kind of strucuture, is to try to use Gaussian mixtures because Gaussian mixtures can nicely fit such clustered data.

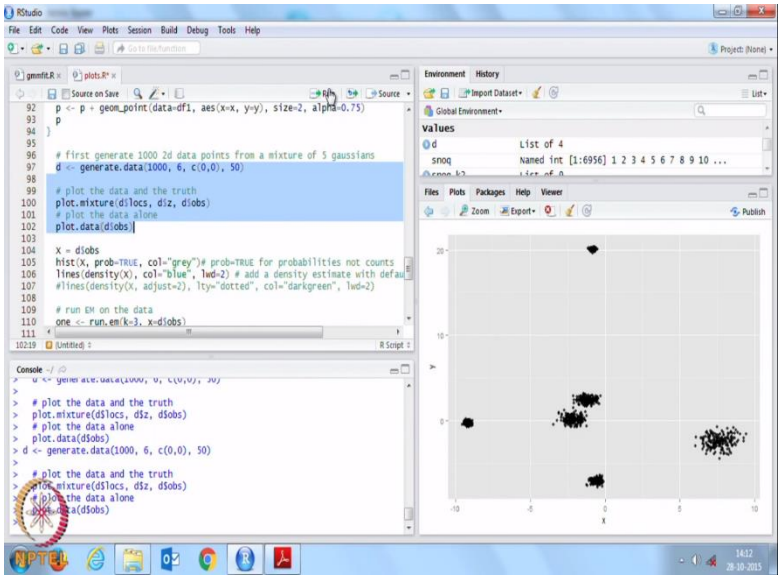
(Refer Slide Time: 07:10)



This is another figure from Bishop's book, which is an illustration of three different Gaussians. As mentioned earlier, these Gaussians need not always be well separated. In this case these Gaussians are overlapping because of the choice of mean and variance. From the subfigure on the left, you know that or you are being told that there are three different Gaussians. However, when you look at the data and try to plot the fitted density, it looks similar to the subfigure in the center. The subfigure on the right is the surface plot of the distribution.

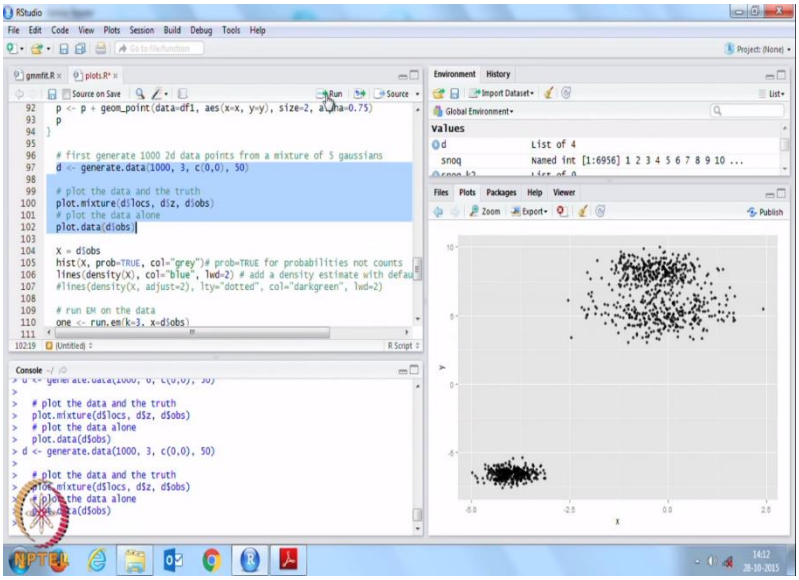
Another thing that you should observe from the subfigure on the left is that it was generated from a three-component Gaussian mixture with weights 0.5, 0.3 and 0.2, which means that the red Gaussian is contributing most of the mass. It is again observed in the subfigure in the center when you plot the density. So the probability at mean of the red gaussian is the highest and then a bit lower for the green Gaussian and the lowest for the blue Gaussian. Let me show you some more examples of these densities.

(Refer Slide Time: 08:47)



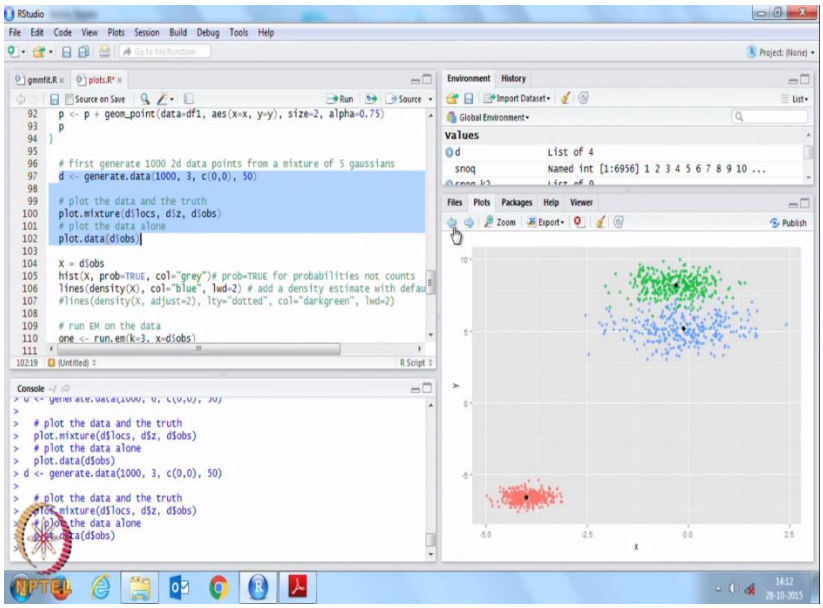
I generated the data shown here from six different components. Let me reduce the number of components to three.

(Refer Slide Time: 09:02)



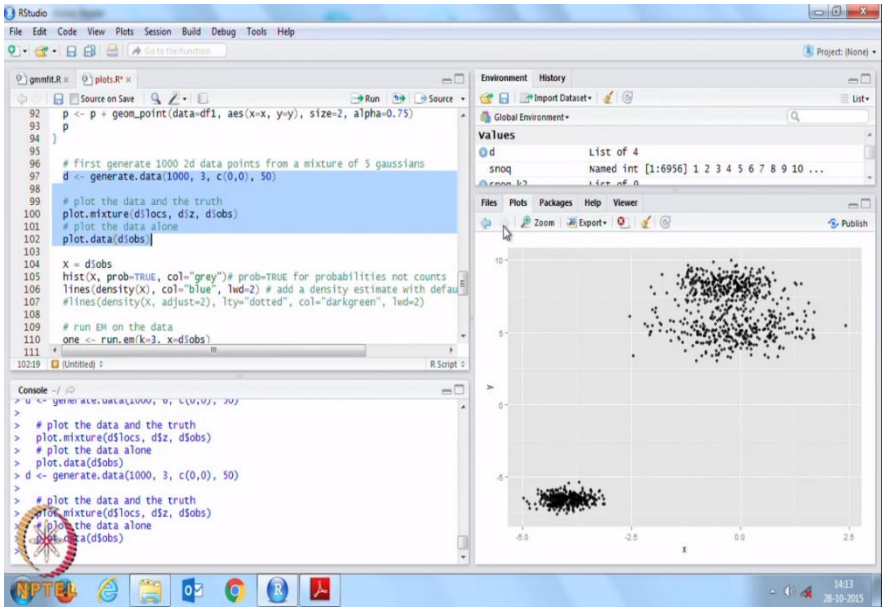
So here there are three components but two among them are highly overlapping.

(Refer Slide Time: 09:02)



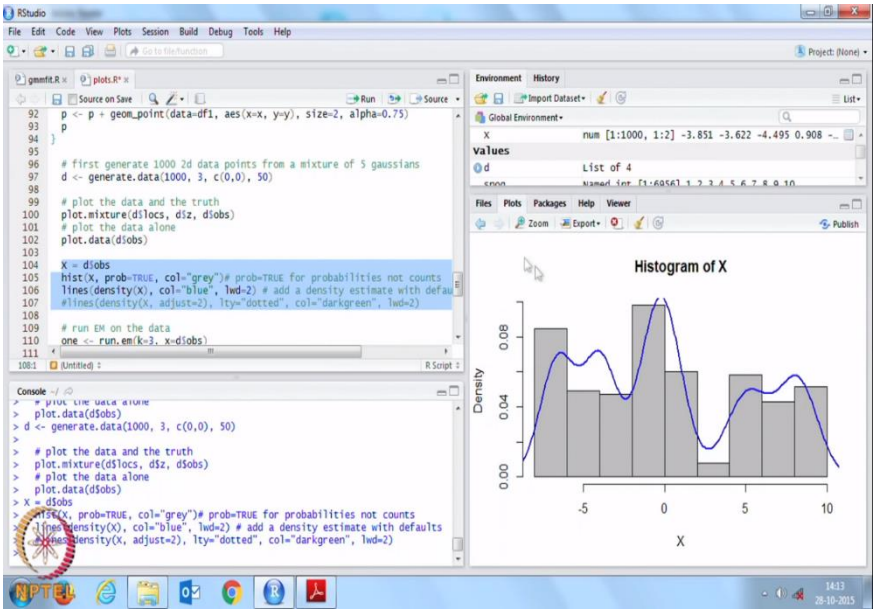
So if you see how the data shown previously was generated, it was generated by three Gaussians as shown here.

(Refer Slide Time: 09:23)



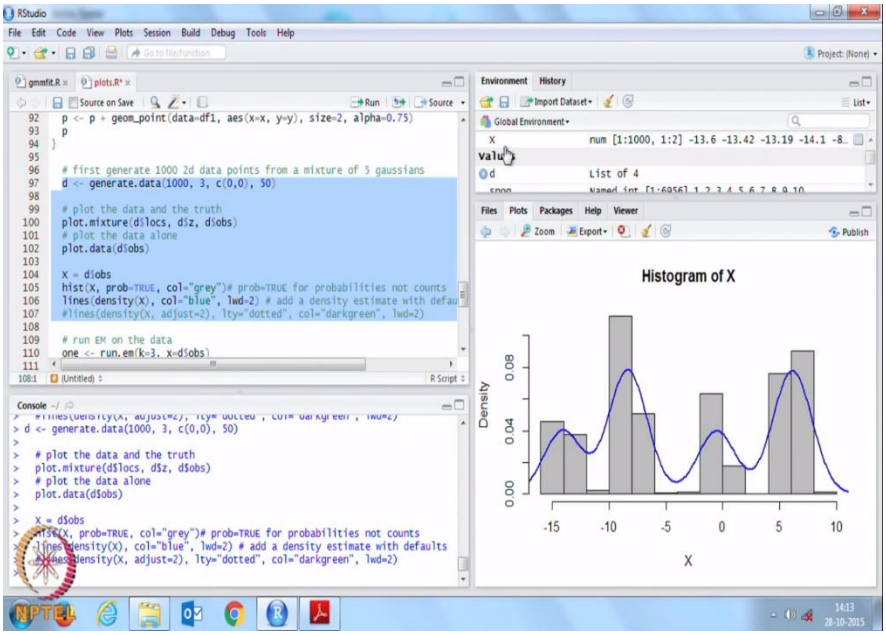
But when you look at the data, you don't know how it was generated. It looks like as shown here. So sometimes it may not be apparent that there are exactly three different components.

(Refer Slide Time: 09:33)

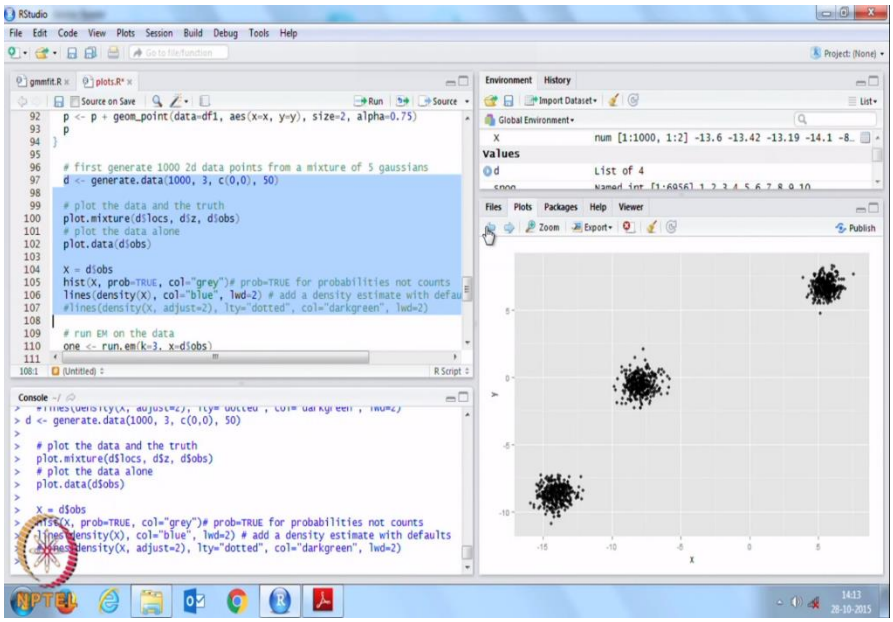


And when you plot the fitted density for the data shown previously, you typically see a density curve as shown here, which has multiple modes. Now this is a fitted density for a three component Gaussian. You see it does not necessarily have exactly three modes. So it depends on the samples that you have. So let us see a few more density plots.

(Refer Slide Time: 10:00)

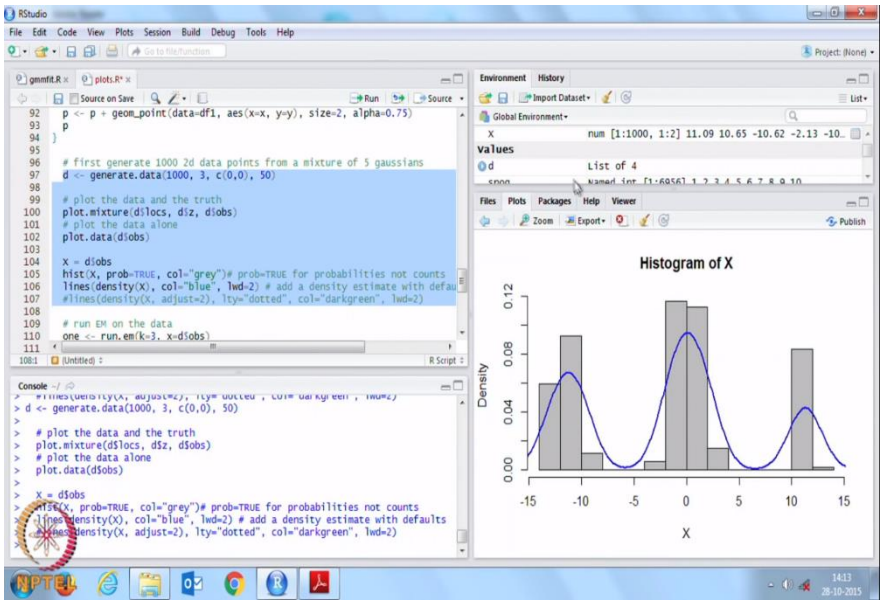


(Refer Slide Time: 10:05)

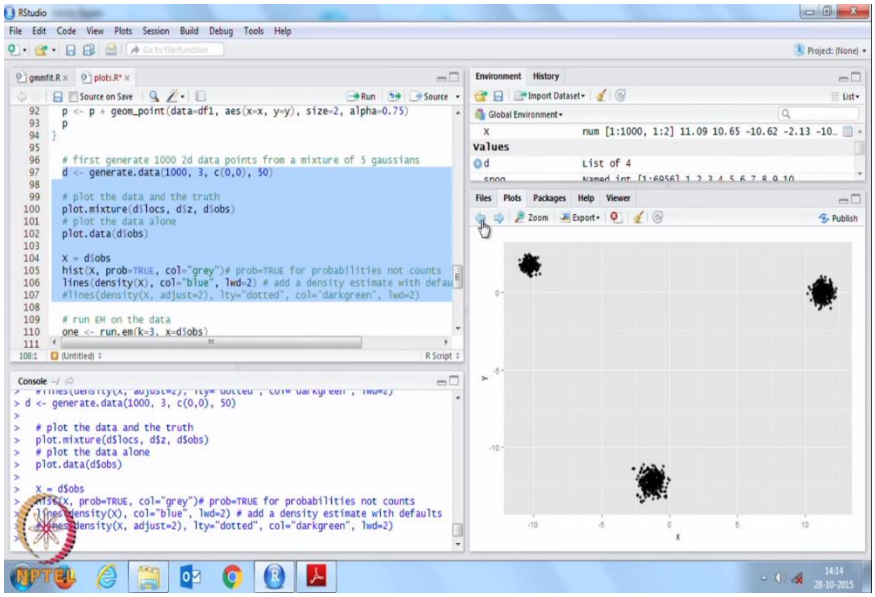


The data that I generated is shown here and the fitted density is shown in the previous figure. The fitted density has four modes.

(Refer Slide Time: 10:12)

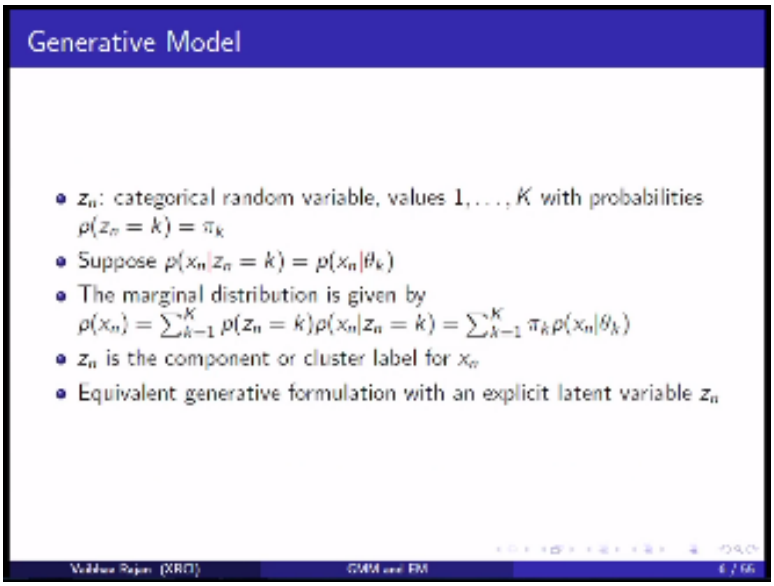


(Refer Slide Time: 10:16)



Upon running again, I generate the data shown here. It has exactly three modes as shown in the previous figure. The data is very well separated and forms three clusters.

(Refer Slide Time: 10:28)



We can also formulate equation (1) as a generative model for selecting a component. Once you select a component, you have selected the Gaussian corresponding to that component. As you know the parameters of the selected Gaussian, you then use them to sample data from that Gaussian. So that would be a generative model for a Gaussian mixture.

To make it more formal, let us take z_n to be a categorical random variable which takes values from 1 to K with the probability of z_n equal to k being exactly equal to π_k.

$$p(z_n = k) = \pi_k$$

Suppose that the probability of the data x_n given $z_n = k$ is the probability of x_n given that you know the parameters for that particular component. I express it as:

$$p(x_n|z_n = k) = p(x_n|\theta_k),$$

where θ_k represents the parameters of the kth component.

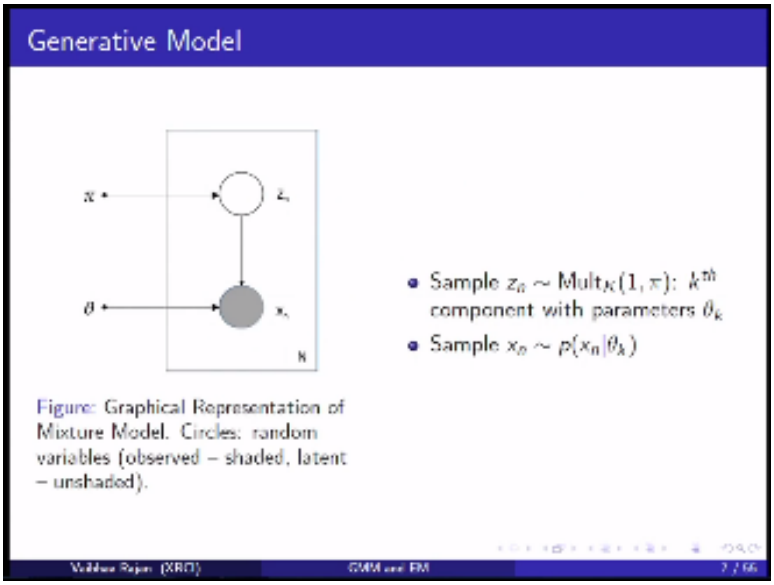
So the marginal distribution can be expressed as the probability of $z_n = k$ (i.e., you select the component) times the probability of x_n given $z_n = k$ (i.e., the probability of x_n coming from

exactly that component). By what we have assumed, the first probability is π_k and the second probability is probability of x_n given θ_k.

$$p(x_n) = \sum_{k=1}^{k} p(z_n = k)\, p(x_n | z_n = k) = \sum_{k=1}^{K} \pi_k\, p(x_n | \theta_k) \qquad (2)$$

So this is an equivalent generative formula with an explicit latent variable z_n.

(Refer Slide Time: 12:24)



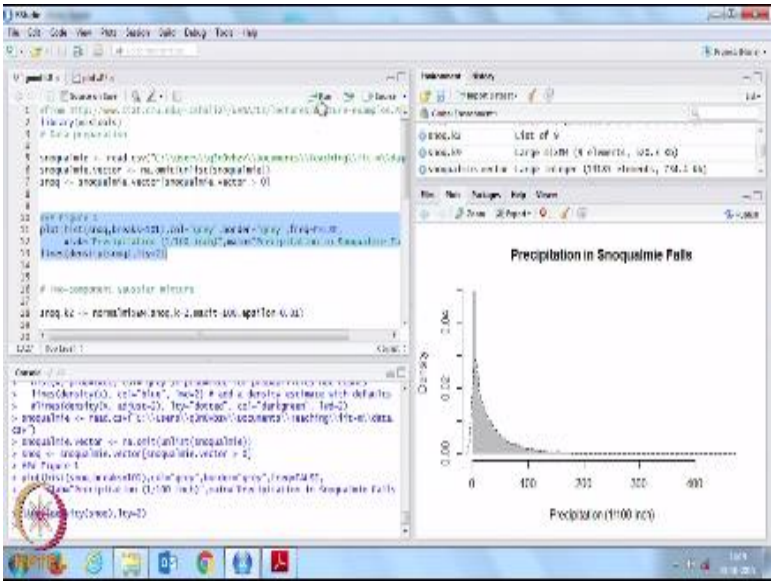
We can represent this graphically. The shaded circle shown here represents the observed data x_n. The unshaded circle represents the latent variable z_n. The latent variable is governed by the parameter π. Once you know the latent variable z_n, the observed data x_n is generated by using the known parameters from θ corresponding to that latent variable z_n. So if you have to generate data from such a model, you first sample z_n from the categorical distribution, which just a special form of multinomial distribution with parameter 1.

And once you samples z_n, you get the k^{th} component with parameters θ_k and then you sample x_n from that probability distribution. So in our case the probability distribution is a Gaussian but it need not be Gaussian. It could be exponential or any complicated probability distribution.

We are assuming that there is a joint distribution and that the marginal distribution representing the probability of x_n can be written as shown in equation (2). For each of the k components, we take the probability of $z_n = k$ and then the probability of x_n being generated from that component. So this is exactly what the graphical model shown here represents. If we write down the probability distribution represented by this graphical model, it will be exactly as shown in equation (2), except that there is a summation term that takes care of all the different components.

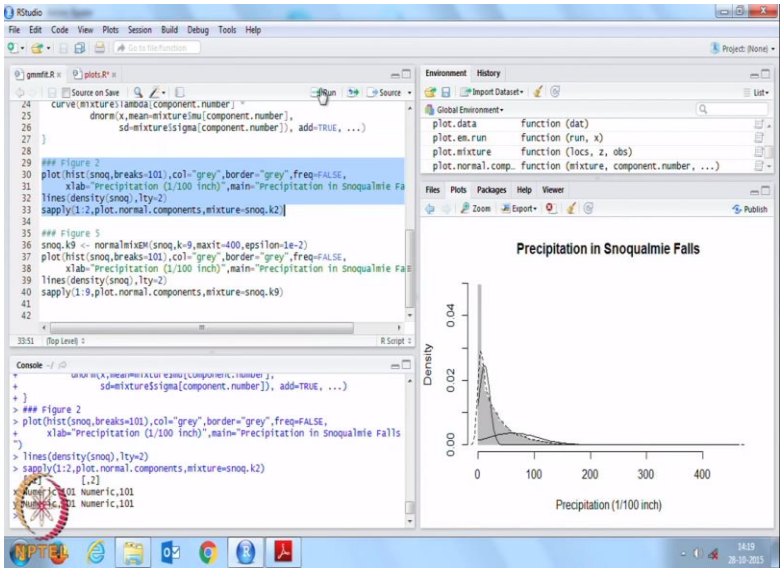
But for a single component, it is the term inside the summation in equation (2). We are choosing a component and then sampling the data point given the distribution parameters for that component. So remember that the mixture model is just another distribution. One use of it as demonstrated earlier is to model clustered data but you can also model other kinds of data with it. As an example let's look at another data set from the web.

(Refer Slide Time: 15:25)



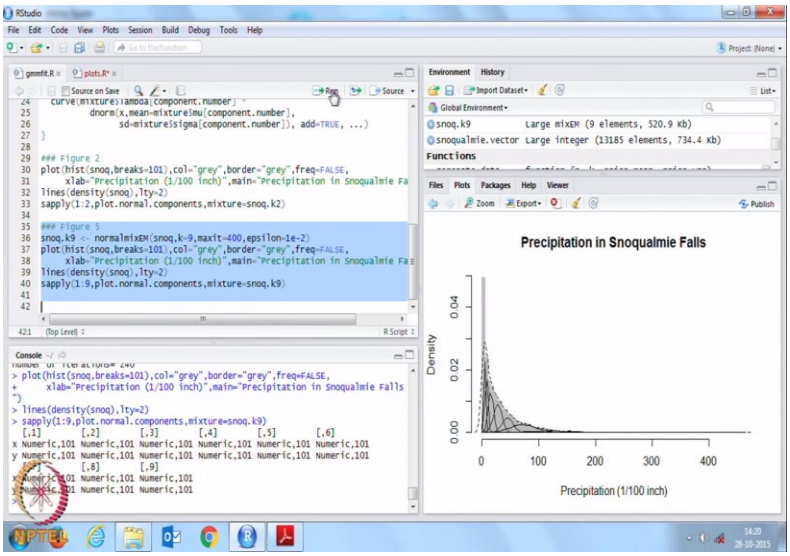
So shown here is the fitted density curve for a data set which records precipitation in the Snoqualmie falls. Suppose we want to model this density, we can model this with a Gaussian mixture. Let's see how it looks when we do it with a Gaussian mixture with two components.

(Refer Slide Time: 16:07)



So we are trying to model the data with a Gaussian mixture. We use a Gaussian mixture with two Gaussians as shown in the figure. The Gaussian mixture with two components is unable to the model the data well. However, we can increase the number of components.

(Refer Slide Time: 16:28)

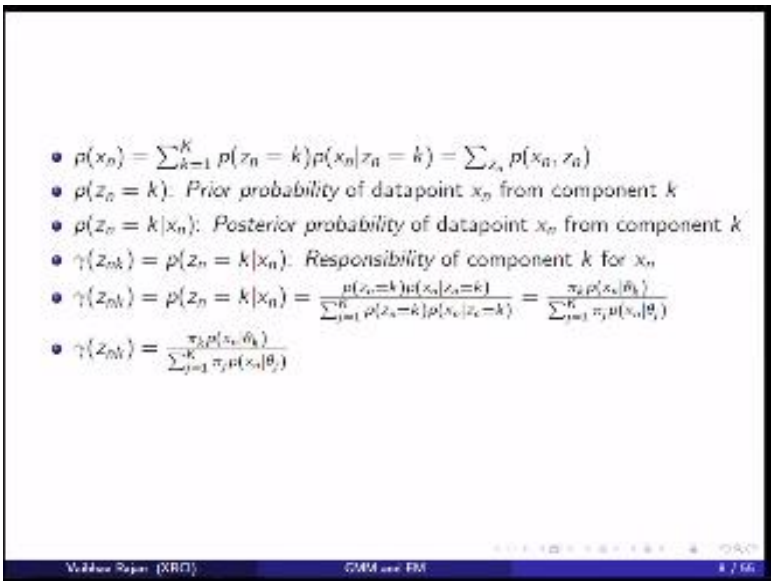


When use use nine components, the fitted distribution gets closer to the true distribution of the data as shown here. So a Gaussian mixture with nine components, in which each of the Gaussians have different mean and covariance parameters, models the data reasonably well.

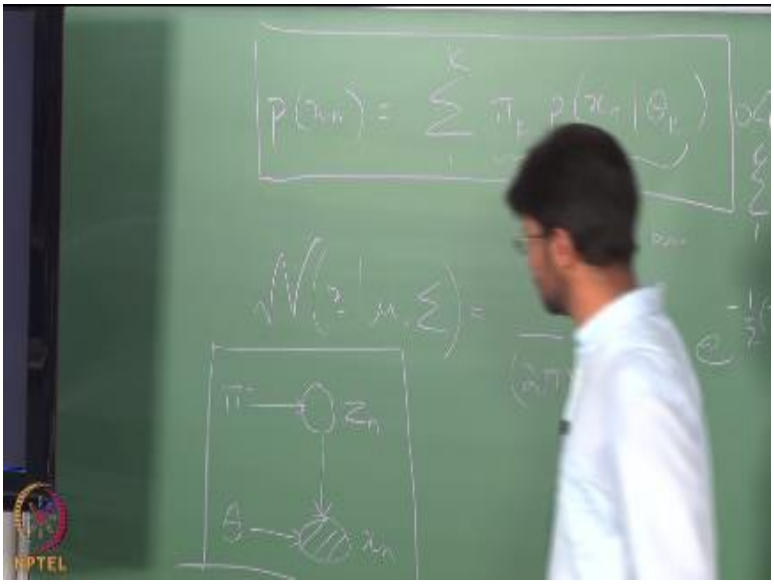
So Gaussian mixture by that way is very versatile. It can model a lot of different distributions by just choosing the right number of components and choosing the parameters appropriately. So it models not just clustered structure.

When we are fitting the model, we want to estimate what the right number of components is and also estimate what the parameters corresponding these components are. So if you are just given the data, we don't know what those parameters are. The bulk of the lecture talks about the estimation of these parameters.

(Refer Slide Time: 17:47)



(Refer Slide Time: 18:03)



It would be good if you keep the graphical representation of the mixture model discussed above (and also shown here) in your mind as we go through the lecture because all the maths that we will see will sort of start making sense when you have this model in your mind. So when I talk about some formula like the one in equation (2), the graphical representation can help us see that it is just the probability of x_n being generated.

The generative model is usually more easy to think with. So the generative model would be that I choose the component z_n and then once I choose the component, I choose the corresponding parameters from θ and generate the data point x_n.

If you are doing a clustering task, these components are the cluster labels. So suppose you have three clusters and you want the cluster labels, if you fit a Gaussian mixture model there, the component values 1,2 and 3 would just become the cluster labels in that case. So this would be a probabilistic way of doing clustering.

The probability of $z_n = k$ (i.e., the probability of z_n, the latent variable or the cluster label taking a particular value k) is the prior probability of the data point x_n coming from the component k.

$$p(z_n = k): Prior\ probability\ of\ datapoint\ x_n\ from\ component\ k$$

Now suppose you are given some data set like the "Precipitation in Snoqualmie falls" data set, and you are asked to find what is the label z_n for the corresponding data point x_n.

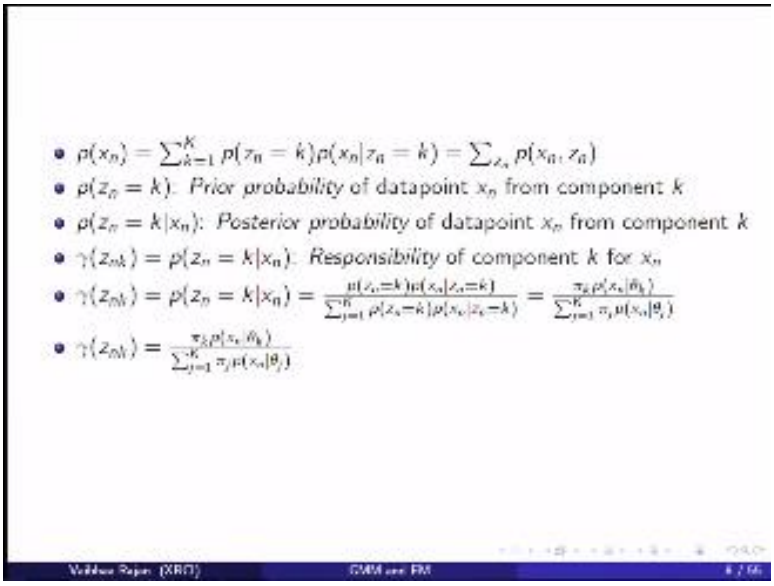
(Refer Slide Time: 20:34)

Suppose you have the data points as show here, which have two clusters. If you knew how the data points were generated, then we would able to assign the data points within the dotted circle above to the same cluster (say having cluster label 1) and assign the other points within the dotted circle below to the another cluster (say having cluster label 2). However, you do not know how the data was generated.

So once you are given this data, you have to infer what these parameters (π and θ) are and given that you are using the mixture model to fit the data, you have to infer the z_n value for each of the data points.

So the z_n values for all the data points in one component will have the same value (say $z_n = 1$) and the z_n values for all the other data points in the other component will have the other value (say $z_n = 2$). Of course, for clustering the cluster labels can be interchanged among the components.

(Refer Slide Time: 21:47)



So this probability, the posterior probability of the data point x n coming from component k is so important that it is given a name of its own. It's called the responsibility. I am going to write that

down as well because we will you reuse it again and again and again. Oh, he asked me not to use this.

(Refer Slide Time: 22:14)

$$\gamma_{nk} = p(z_n = k|x_n)$$

The equation represents the posterior probability of $z_n = k$ given the data. It represents the responsibility of component k for data point x_n.

So until now, I have described it in a way that, there are these different components and only one of these components is responsible for giving rise to a particular data point. However, that is from the generative point of view. But if you look at it probabilistically, each of the components is contributing something towards the probability of that data point and the weight of that contribution from component k is given by π_k.

So your clustering need not always be a hard clustering (of this component versus that component), it can be soft clustering as well, where the cluster label can be probabilistic. For example, it can be cluster label 1 with probability 0.5, cluster label 2 with probability 0.3 and so on. So when you

express it as a probability, i.e., $p(z_n = k|x_n)$, then you get the option of doing both hard clustering as well as soft clustering.

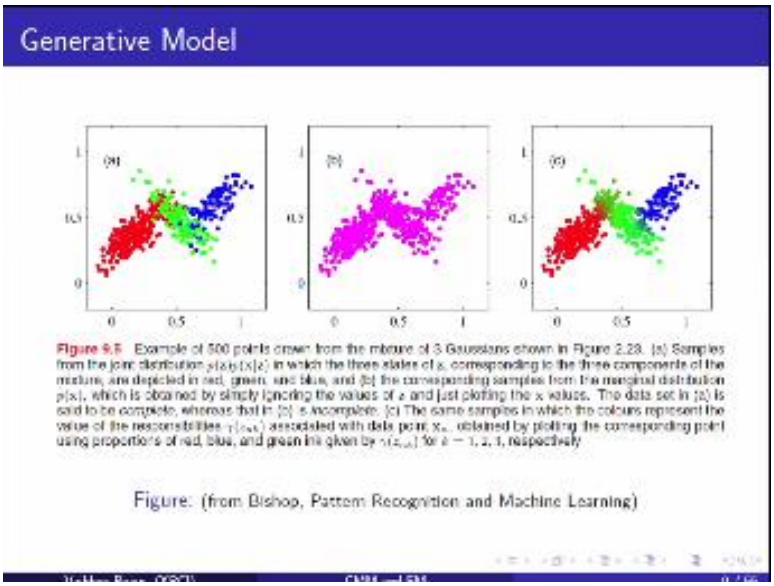
So now you can use Bayes rule to get derive the formula for responsibility.

$$\gamma(z_{nk}) = p(z_n = k|x_n) = \frac{p(z_n = k)\, p(x_n|z_n = k)}{\sum_{j=1}^{K} p(z_n = j)\, p(x_n|z_n = j)} = \frac{\pi_k p(x_n|\theta_k)}{\sum_{j=1}^{K} \pi_j\, p(x_n|\theta_j)}$$

So the equation is straightforward. When you substitute for the prior $p(z_n = k)$ in the LHS, you get π_k in the RHS. Also, the probability of x_n given that you know the component (i.e., $p(x_n|z_n = k)$) in the LHS can be substitued with $p(x_n|\theta_k)$, where θ_k are the parameters for k_{th} component.

Observe that you don't know the responsibility values, $\gamma(z_{nk})$, until you know all the parameters. In addition to the value of k, you need to know all the π_ks and θ_ks for all the k components, which in the case of Gaussian is all the μ_ks and Σ_ks.

(Refer Slide Time: 25:26)



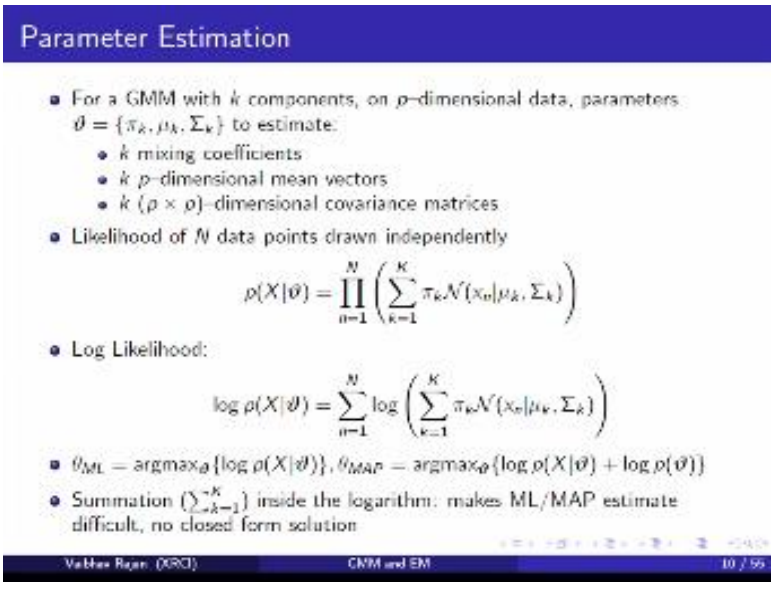
So here's another very interesting picture from Bishop's book. The data was generated from three different Gaussians - the red, green and blue Gaussians. However, you don't know how the data

was generated and when you see the data, you see something similar to that shown in the subfigure in the center. Then you try to fit a mixture model with three Gaussians on to the data, and plot the repsonsibilities for each of data points as shown in the subfigure on the right.

So the data points plotted with pure red colour have been given the responsibility corresponding to component 1 (component red). The second component is responsible for all the data points plotted in pure blue colour, and component 3 (green component) is responsible for all the data points plotted in pure green color. However, in the border between these components, the data points are plotted with a mixture of green and blue (or red and green) colours depending on probabilities values (or reponsibility values) for the data point corresponding to each of the components. These data points are not completely assigned to one single component (or cluster). This is an example of soft clustering.

You can see that there are mistakes in inference because if you do something like a maximum likelihood estimates, these will be most likely from a single Gaussian, and these blue points and these red points will not be identified correctly.

(Refer Slide Time: 27:10)



So suppose you are given data, and you want to fit a Gaussian mixture model to it, to either infer the clusters or to just fit the density. For either case you need to estimate the parameters of the model. If you have p-dimensional data and are using a Gaussian mixture with k components to fit the data, we need to find out the k mixing coefficients ($\pi_k s$), the k mean parameter vectors ($\mu_k s$) each of which is p-dimensional and the k covariance matrices (Σ_ks), corresponding to each of the k components. For now, we assume that we know the value of k (i.e., the number of components to use) and will later come back to discuss how k is estimated.

By observing the dimension of the covariance matrices, Σ_ks, we can infer that fitting the model is going to get difficult for high dimensional data because the dimension of the covariance matrices scale quadratically with the number of dimensions in the data. Dimension of the covariance matrix is $p \times p$, where p is the dimensionality of the data.

To estimate the parameters of the model, we will take the standard route of maximum likelihood. The likelihood of N data points drawn independently, for the case of Gaussian mixture model with k components is given by:

$$p(X|V) = \prod_{\mathrm{n}=1}^{\mathrm{N}} \left(\sum_{k=1}^{K} \pi_k \, N(x_n|\mu_k, \Sigma_k) \right) \tag{3}$$

where, $\vartheta = (\pi,\ \mu,\ \Sigma)$ are the parameters of the model, and $N(x_n|\mu_k, \Sigma_k)$ is the probability for data point x_n sampled from the Gaussian distribution corresponding to the k^{th} component.

We apply logarithm to both sides of equation (3), which converts the outer product to a summation.

$$\log p(X|\vartheta) = \sum_{\mathrm{n}=1}^{\mathrm{N}} \log \left(\sum_{k=1}^{K} \pi_k \, N(x_n|\mu_k, \Sigma_k) \right) \tag{4}$$

The summation within the logarithmic term in the log likelihood computation shown in equation (4) causes lot of problems in the estimation of the parameters of a Gaussian mixture.

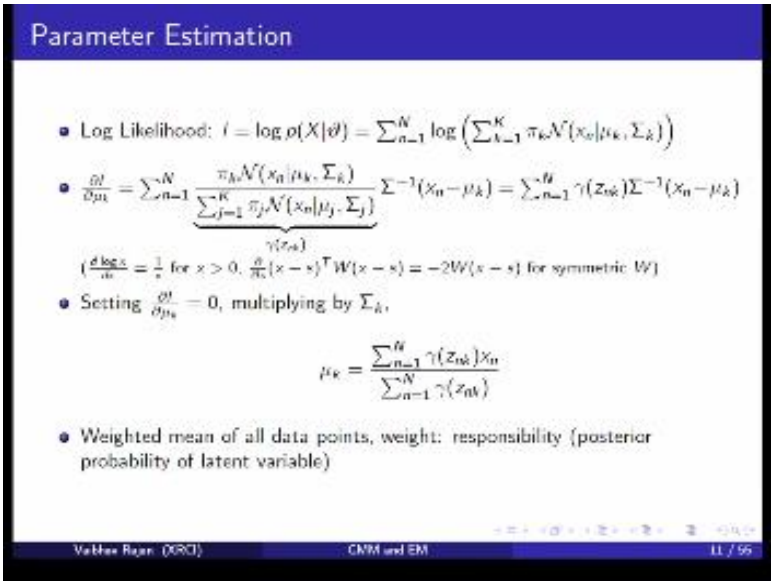
Since our objective is maximize the likelihood (and thereby also maximize the log likelihood) of the data X, we should be able to estimate the parameters ϑ, by differentiating equation (4) w.r.t. to the each of the parameters in ϑ, and equating them to 0.

$$\frac{\partial \log p(X|\vartheta)}{\partial \boldsymbol{\mu_k}} = \mathbf{0}, \qquad \frac{\partial \log p(X|v)}{\partial \boldsymbol{\Sigma_k}}, \qquad \frac{\partial \log p(X|\vartheta)}{\partial \boldsymbol{\pi_k}} = \mathbf{0}$$

for all the values of k.

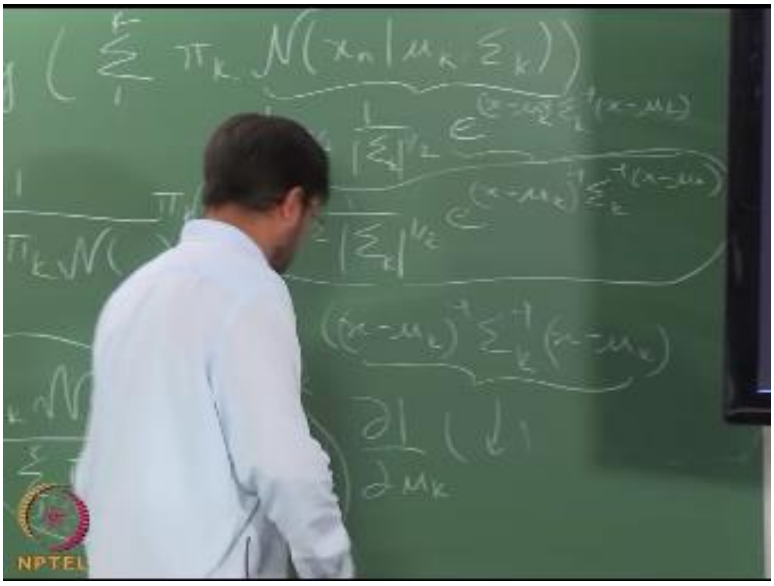
However, the summation within the logarithm term in the expansion of $\log \mathrm{p}(X|\vartheta)$, shown in equation (4), poses problems in differentiating $\log p(X|\vartheta)$ w.r.t. to each of the parameters in ϑ and we are not going to get a closed-form solution. This is one of the main problems for estimating the parameters of a Gaussian mixture.

(Refer Slide Time: 30:13)



However, let's attempt to estimate the parameters of the model by differerntiating the log-likelihood with respect to each of the parameters of the model and equating them to 0. We will make one crucial assumption which is that we know the responsibility terms γ_{nk}, for each of the data points x_n, for each of the k components. So let us start with the parameter μ_k and see what we get by doing some algebra.

(Refer Slide Time: 31:04)



The log-likelihood is given by:

$$l = \sum_{n=1}^{N} \log\left(\sum_{k=1}^{K} \pi_k \, N(x_n|\mu_k, \Sigma_k)\right) \tag{5}$$

where π_k (not bolded) in the equation of log-likelihood refers to mixing coefficients of the mixture model.

Also, we know that the probability distribution of a Gaussian can be written as:

$$N(x \mid \mu_k, \Sigma_k) = \frac{1}{(2\boldsymbol{\pi})^{\frac{p}{2}}} \frac{1}{|\Sigma_k|^{\frac{1}{2}}} \, e^{-\frac{1}{2}(x-\mu_k)^T \, \Sigma_k^{-1} \, (x-\mu_k)} \tag{6}$$

Where, $\boldsymbol{\pi}$ (bolded) in the equation refers to the constant $\boldsymbol{\pi}$ (defined as the ratio of circle's circumference to its diameter).

Substituting equation (6) in equation (5), the log-likelihood can be rewritten as:

$$l = \sum_{n=1}^{N} \log\left(\sum_{k=1}^{K} \pi_k \left(\frac{1}{(2\boldsymbol{\pi})^{\frac{p}{2}}} \frac{1}{|\Sigma_{\mathrm{k}}|^{\frac{1}{2}}} e^{-\frac{1}{2}(x_n-\mu_k)^T \Sigma_{\mathrm{k}}^{-1} (x_n-\mu_k)}\right)\right) \tag{7}$$

We now find the partial derivatives of the log-likelihood with respect to parameter μ_k:

$$\frac{\partial l}{\partial \mu_k} = \sum_{n=1}^{N} \frac{1}{\sum_{j=1}^{K} \pi_j \, N\left(x_n \middle| \mu_j, \Sigma_j\right)} \pi_k \left(\frac{1}{(2\boldsymbol{\pi})^{\frac{p}{2}}} \frac{1}{|\Sigma_{\mathrm{k}}|^{\frac{1}{2}}} e^{-\frac{1}{2}(x_n-\mu_k)^T \Sigma_{\mathrm{k}}^{-1} (x_n-\mu_k)}\right)$$

$$\cdot \frac{\partial(-\frac{1}{2}(x_n-\mu_k)^T \Sigma_{\mathrm{k}}^{-1} (x_n-\mu_k))}{\partial \mu_k}$$

$$= \sum_{n=1}^{N} \left(\frac{\pi_k \, N(x_n|\mu_k, \Sigma_k)}{\sum_{j=1}^{K} \pi_j \, N\left(x_n \middle| \mu_j, \Sigma_j\right)}\right) \frac{\partial(-\frac{1}{2}(x_n-\mu_k)^T \Sigma_{\mathrm{k}}^{-1} (x_n-\mu_k))}{\partial \mu_k} \tag{8}$$

However, we know that the responsibility of the k^{th} component for the data point x_n can be written as:

$$\gamma_{nk} = \frac{\pi_k \, N(x_n|\mu_k, \Sigma_k)}{\sum_{j=1}^{K} \pi_j \, N\left(x_n \middle| \mu_j, \Sigma_j\right)} \tag{9}$$

Substituting equation (9) in equation (8), we get:

$$\frac{\partial l}{\partial \mu_k} = \sum_{n=1}^{N} \gamma_{nk} \left(-\frac{1}{2} \frac{\partial\left((x_n-\mu_k)^T \Sigma_{\mathrm{k}}^{-1} (x_n-\mu_k)\right)}{\partial \mu_k}\right) \tag{10}$$

From the matrix cook book which contains complex matrix derivatives, we have:

$$\frac{\partial}{\partial s}(x-s)^T W(x-s) = -2W(x-s), for\ symmetric\ W \tag{11}$$

Using equation (11), we have that:

$$\frac{\partial}{\partial \mu_k}(x_n-\mu_k)^T \Sigma_{\mathrm{k}}^{-1}(x_n-\mu_k) = -2\Sigma_{\mathrm{k}}^{-1}(x_n-\mu_k) \tag{12}$$

(because covariance matrix Σ and thereby Σ^{-1} are symmetric matrices)

Substituting equation (12) in equation (10), we get:

$$\frac{\partial l}{\partial \mu_k} = \sum_{n=1}^{N} \gamma_{nk} \left(-\frac{1}{2}\left(-2\Sigma_k^{-1}(x_n-\mu_k)\right)\right) = \sum_{1}^{n} \gamma_{nk}\ \Sigma_k^{-1}(x_n-\mu_k) \tag{13}$$

Equating $\frac{\partial l}{\partial \mu_k} = 0$ from equation (13) and multiplying by Σ_k, we get:

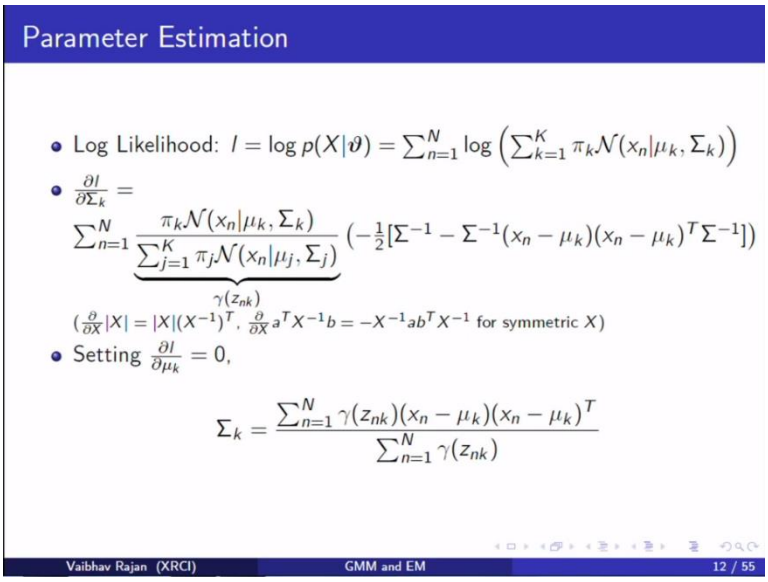
$$\Sigma_k \frac{\partial l}{\partial \mu_k} = \Sigma_k \left(\Sigma_k^{-1} \sum_{n=1}^{N} \gamma_{nk}\ (x_n-\mu_k)\right) = \Sigma_k 0 = 0$$

$$\mu_k = \frac{\Sigma_{n=1}^{N} \gamma_{nk} x_n}{\Sigma_{n=1}^{N} \gamma_{nk}} \tag{14}$$

It can be observed from equation (14) that the mean inferred for the k^{th} component Gaussian, μ_k, is the weighted mean of all the data points, where the weight is the responsibility of that cluster towards that data points.

It is important to note that we do not know the responsibilities γ_{nk}. We have assumed that we know the responsibilities, substituted it and did the math to arrive at a nice form for μ_k, shown in equation (14). However, the responsibility γ_{nk} has all the unknowns inside it, including mixing coefficient π_k, mean parameter μ_k and covariance parameter Σ_k, as shown in equation (9).

(Refer Slide Time: 37:41)



Similarly, we can arrive at a nice form for Σ_k by derivating the log-likelihood of the data w.r.t. to Σ_k and equating it to 0. Here also we assume that we know the responsibilities terms.

$$\frac{\partial l}{\partial \Sigma_k} = \sum_{n=1}^{N} \underbrace{\frac{\pi_k N(x_n|\mu_k, \Sigma_k)}{\sum_{j=1}^{K} \pi_j N(x_n|\mu_j, \Sigma_j)}}_{\gamma_{nk}} \left(-\frac{1}{2}[\Sigma^{-1} - \Sigma^{-1}(x_n - \mu_k)(x_n - \mu_k)^T \Sigma^{-1}]\right)$$

$$(\frac{\partial}{\partial X}|X| = |X|(X^{-1})^T, \frac{\partial}{\partial X} a^T X^{-1} b = -X^{-1} a b^T X^{-1} \; for \; symmetric \; X)$$

Setting $\frac{\partial l}{\partial \Sigma_k} = 0$,

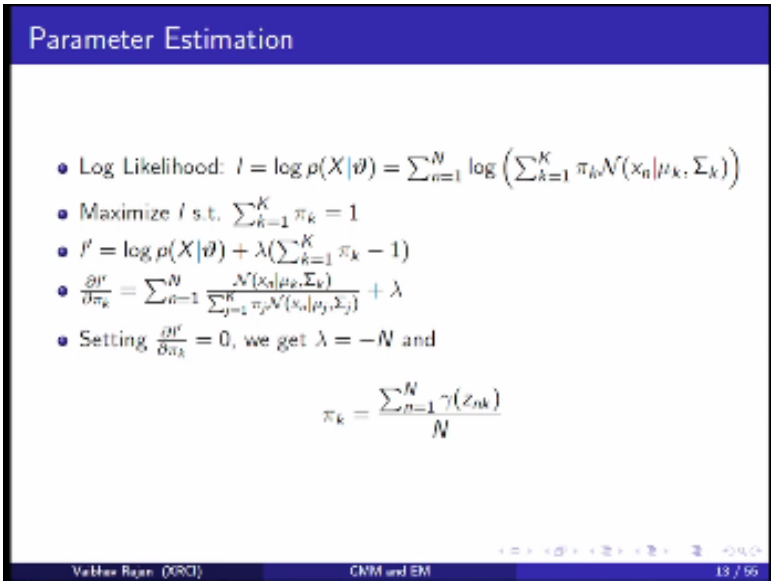
$$\Sigma_k = \frac{\sum_{n=1}^{N} \gamma_{nk}(x_n - \mu_k)(x_n - \mu_k)^T}{\sum_{n=1}^{N} \gamma_{nk}}$$

Responsibility is the posterior probability of the latent variable given that you have the data point. So essentially it's trying to capture which of the components generated that data point.

So right now we do not really have anything. We currently do not have the mean parameters or the covariance parameters. We just know that if we knew the responsibilities, which we don't

know, we could get a nice form for mean parameters μ_ks and covariance parameters Σ_ks. However without this assumption, the derivations for μ_ks and Σ_ks does not hold.

(Refer Slide Time: 39:17)



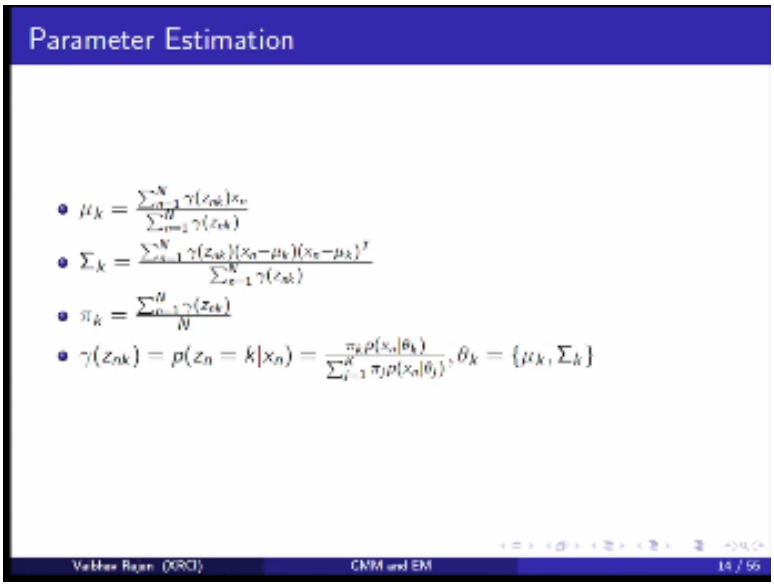
When we take the derivatives with respect to π_k, we have to be careful. We have to make sure that we use the constraint $\sum_{k=1}^{K} \pi_k = 1$. So we cannot directly differentiate the log-likelihood with respect to π_k. We have to use Lagrange multipliers.

So we just take a Lagrange multiplier here, take this constraint, and do the differentiation. The responsibility terms crops up in the derivative. If we set the derivative to 0, we can get langrange multiplier $\lambda = -N$ and we also get the value of π_k as:

$$\pi_k = \frac{\sum_{n=1}^{N} \lambda_{nk}}{N}$$

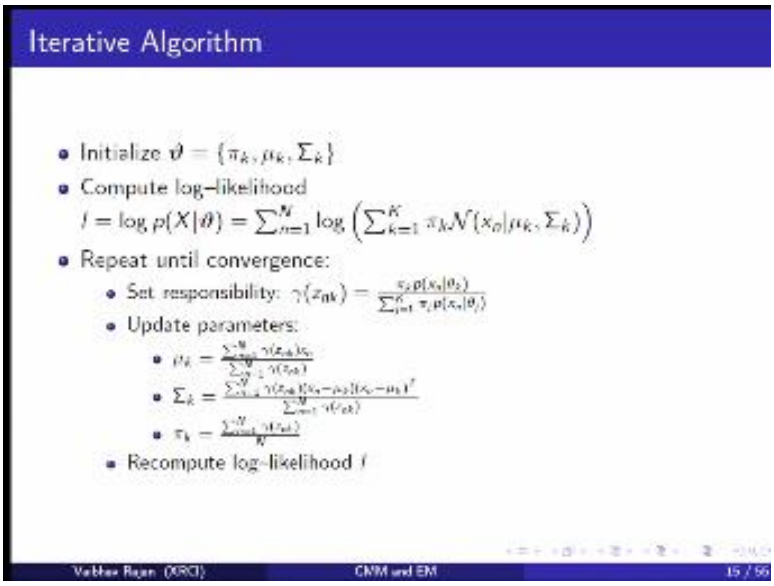
So if you take the sum of all the responsibilities over all data points over all components, it is equal to N. So the kth mixture weight π_k is nothing but the proportion of the responsibilities that are coming from the kth component towards all the data points, and you are taking the proportion that is given by those responsibilities with respect to all of it, i.e., the sum of it.

(Refer Slide Time: 40:53)



So let's summarize. What we found is that we have very nice forms for μ_k, Σ_k and π_k, given that we know the responsibilities, which is the posterior probability of the latent variable coming from the k component for the nth data point. Can you think of how you can use this to create an algorithm for estimating the Gaussian mixture parameters?

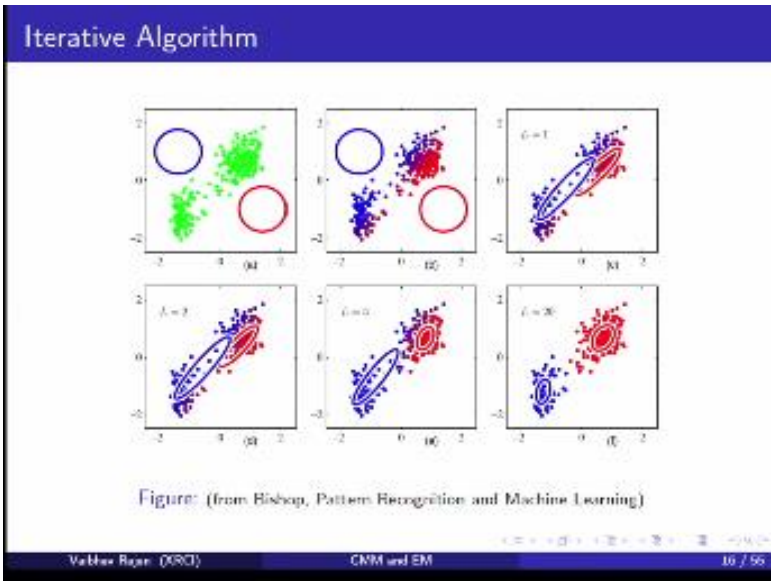
(Refer Slide Time: 41:53)



I will denote by ϑ all the parameters. We start with some guess of the parameters, and then we will compute the log-likelihood. We can compute it because we know or have guessed the parameters. Then, we will set the responsibilities because we know or have guessed all the parameters necessary for computing them. Since given the responsibility we can compute all the parameters again, we will use this to get a new guess, and that way we will iteratively keep refining our guess.

Now, this looks very ad hoc but is actually theoretically quite sound. The reason why this is a good thing to do is we will show that this is guaranteed to increase the likelihood at every iteration. We will see that when we understand how EM works. So this turns out to be actually an instance of the EM algorithm. It is quite intuitive.

(Refer Slide Time: 43:11)



So this is an example of exactly that algorithm. We start with some data. We start with some guesses for the Gaussians, and iteratively we see that the Gaussians converge to nicely fit the data. The parameters that you get here will fit the data very well. The only problem is that it usually takes a long time to come to the right parameters. In this whole iterative algorithm, we are assuming that we know the value of k.

IIT Madras Production

Funded by

Department of Higher Education

Ministry of Human Resource Development

Government of India

www.nptel.ac.in

NPTEL

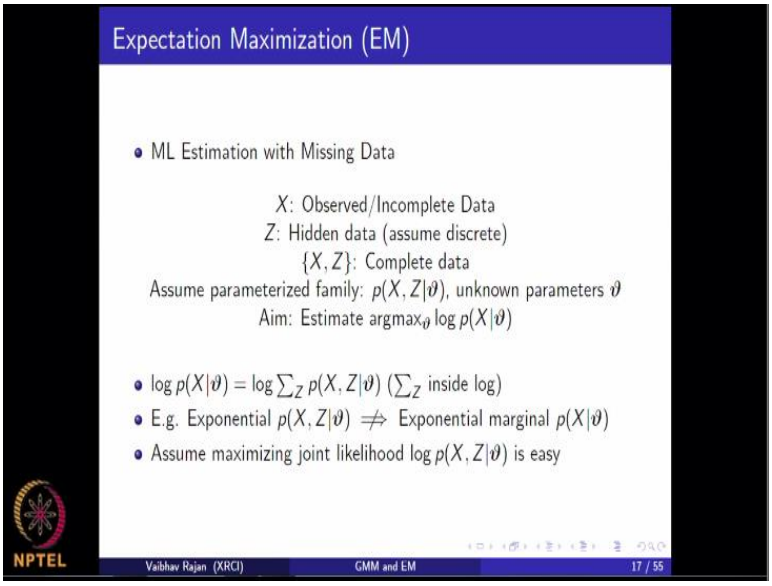
NPTEL ONLINE CERTIFICATION COURSE

Introduction to Machine Learning

Lecture-77
Expectation Maximization

Prof. Balaraman Ravindran
Computer Science and Engineering
Indian Institute of Technology Madras

(Refer Slide Time: 00:15)



Expectation Maximization (EM) is a way to do maximum likelihood estimation. Initially, it was proposed as a way to do maximum likelihood estimation when you have missing data. Suppose you are given data X which is what you observe and is known to be incomplete (i.e., there are some values that are missing), and you want to get the maximum likelihood estimates of the parameters which are unknown.

Here we are assuming two things. We are assuming that there is some parameterized family (i.e., you are doing some parameterized fitting) for which the joint likelihood is easy to compute. So we

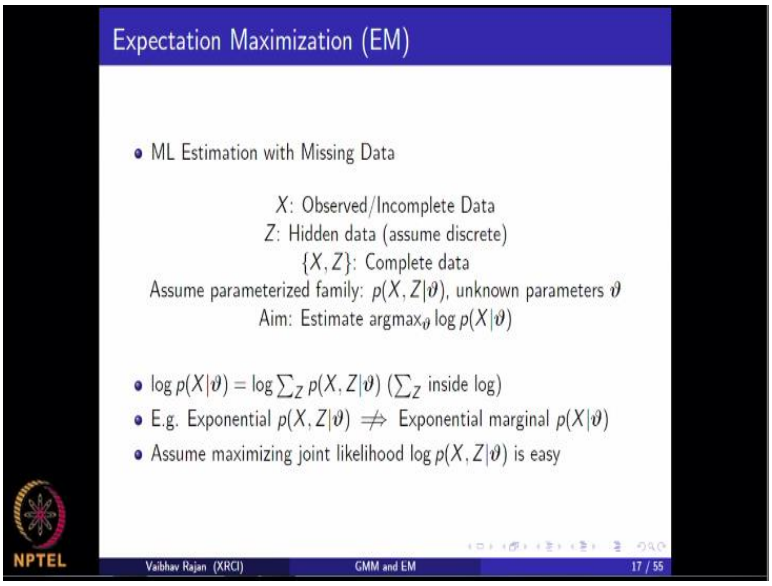
denote by $\{X, Z\}$ the complete data (the observed data plus the hidden data), and we assume some parameterized family from which this data is generated (like a Gaussian or an exponential, and we do not know the unknown parameters which we want to estimate.

So we start seeing connections with what we have seen in the case of Gaussian mixture because if you take the marginal probability here, you again see a summation coming inside the log, and this again poses problems. Also, the marginals need not be of the same family. For example, if the joint probability distribution is an exponential distribution, it does not mean that the corresponding marginal probability distributions also comes from the exponential family.

(Refer Slide Time: 02:04)

EM as it is most commonly used today was proposed by Dempster, Laird and Rubin in their 1977 seminal paper "Maximum Likelihood from incomplete data via the EM algorithm". Even before 1977, a lot of statisticians have developed EM like algorithms, but when we cite EM, we usually cite this paper.

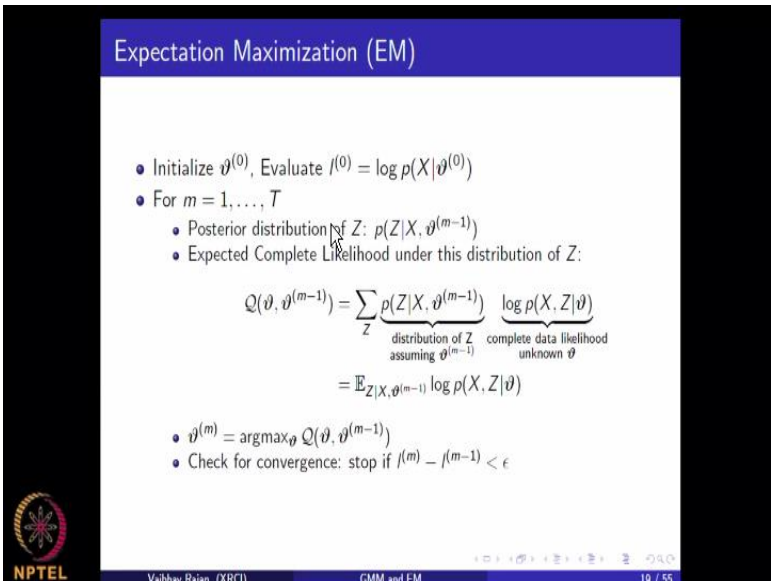
(Refer Slide Time: 02:42)



We have observed or incomplete data and hidden data. For the rest of the discussion, we will assume that this hidden data is discrete, but all the derivations will work if we assume the hidden data to be continuous as well. In which case, we just have to replace the summations with integrals.

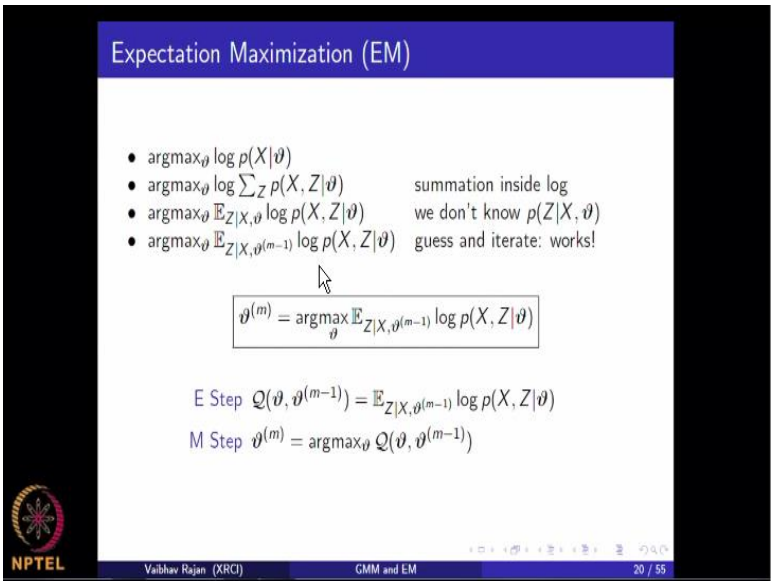
The complete data is the combination of the incomplete data and hidden data. We assume some parameterized family for the complete data, and want to estimate the parameters. This is the problem that EM solves.

(Refer Slide Time: 02:32)



EM is an iterative algorithm, just like what we designed for Gaussian mixture. We again start with a guess of the parameters that we have. We evaluate our first guessed likelihood, and then we iteratively do two steps. First, we compute the posterior distribution of Z given the current estimate of the parameters. After that, we compute the expected complete log likelihood under this distribution, which we'll call as the Q function. Now notice that this, this expectation takes the complete data likelihood. Here the parameters are unknown, and this expectation assumes the distribution of Z given the parameters that we have guessed in the previous round. And then we again get a new guess for the parameters by maximizing this Q function, and taking that argument ϑ which maximizes this Q function.

(Refer Slide Time: 05:08)

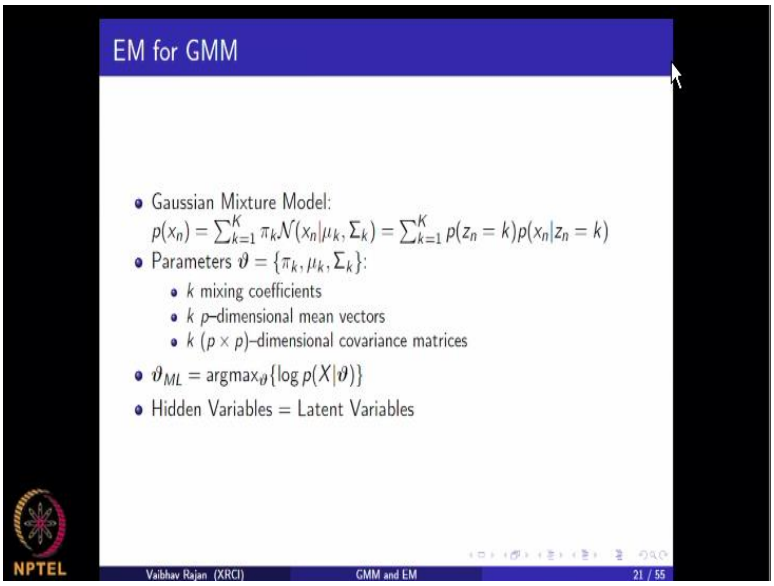


What we wanted was to get the maximum likelihood estimate of ϑ. We see X but X is not complete. X has some missing data Z. So we express the likelihood of observed data X as a marginal distribution of the complete data, and we get into the same problem of the summation being inside the logarithm.

So we decide to not compute the maximum likelihood in this way, but instead compute the maximum taking the expectation of the log likelihood of the complete data under the distribution of Z. But we do not know the real distribution of Z because we do not know the parameters. So we take the guess that we had in the previous round. We compute the expectation of the complete data likelihood under the distribution of Z given the current guess of the parameters.

This works. We will see why it works. So the entire EM algorithm can actually be represented by just this one line. You start with some guess, and then for the next guess you calculate the expectation of the complete data log likelihood under the distribution of the missing data using the previous guess. This can actually be broken down into two steps, where in the E step you compute this expectation, and in the M step you maximize this Q and get the next set of parameters.

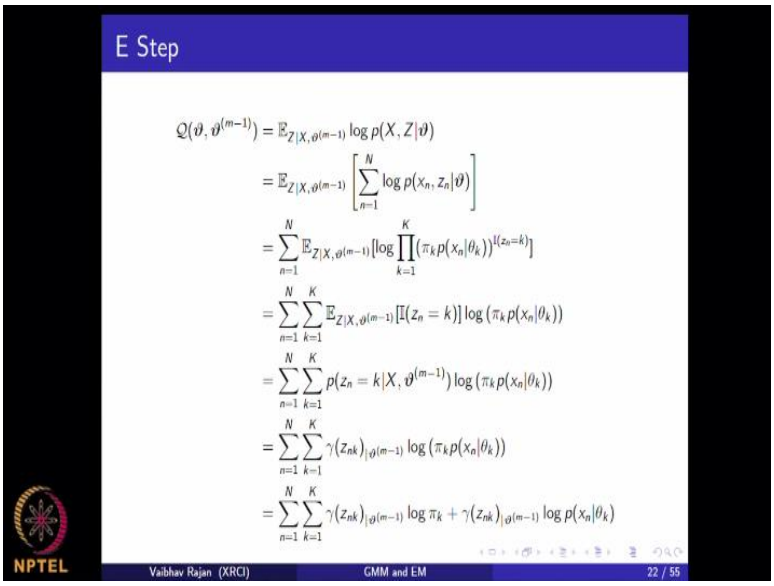
(Refer Slide Time: 07:39)



So let us see how we can get the EM algorithm for Gaussian mixtures. The key thing here is we did not say anything about hidden variables, but the trick here is to use these latent variables as hidden variables. This is how EM is used in a lot of different models, not just Gaussian mixture, but also in a lot of latent variable models. You assume that the latent variables are hidden, i.e., you do not know them, and run the whole EM machinery.

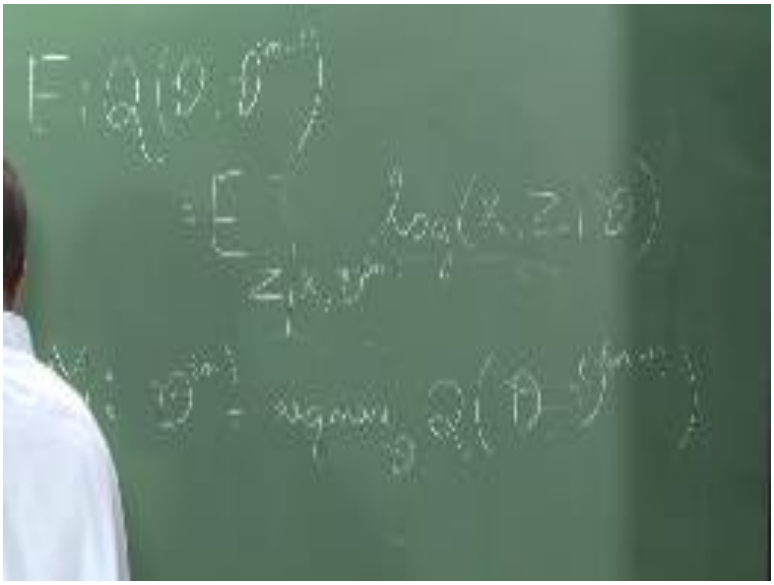
This slides recalls the Gaussian mixture model. We have the Gaussian mixture model, where we want to estimate all the parameters represented by ϑ. We have K components, and each of the component has parameters π_k, μ_k and Σ_k. What we want to find out is the maximum likelihood estimate.

(Refer Slide Time: 08:31)



All right.

(Refer Slide Time: 09:20)



So let's write this down here.

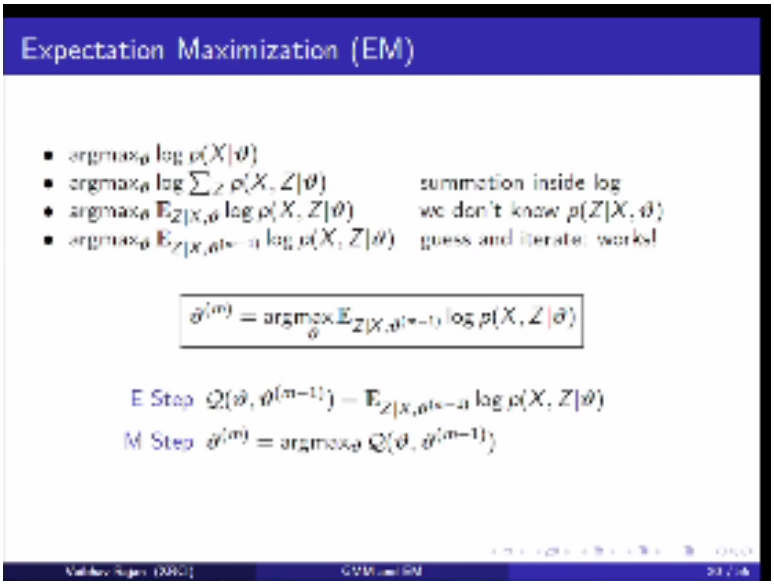
$$E: Q\left(\vartheta, \vartheta^{(m-1)}\right) = E_{Z|X,\vartheta^{(m-1)}} \log(X, Z|\vartheta)$$

The most important thing to remember is that this distribution is taken over the previous guess, where as the expectation is for the complete log-likelihood over the unknown parameters. The M step just gets the next guess.

$$M: \vartheta^{(m)} = argmax_{\vartheta}\, Q(\vartheta, \vartheta^{(m-1)}) \qquad (1)$$

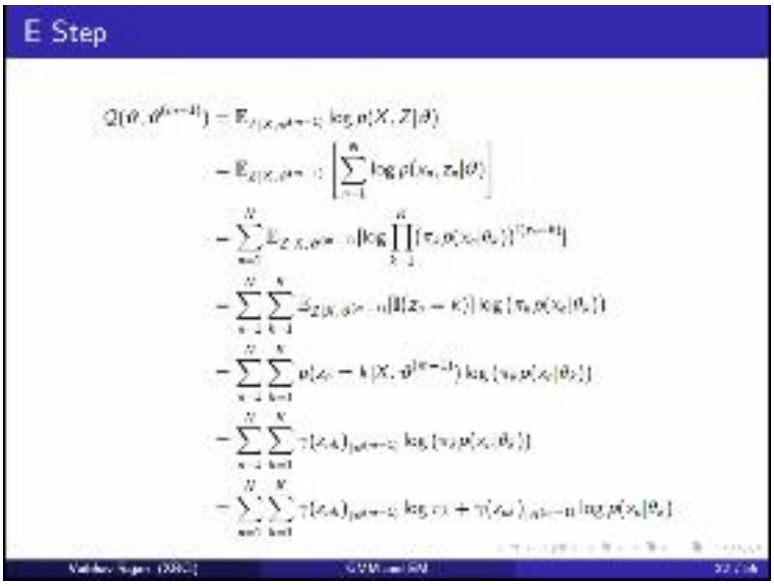
If we want to get the maximum likelihood parameters, we first need to compute this Q function. Then usually the M step is easier. It's just the expectation computation in E step that requires some work. Once you get Q, then the maximization step is just computing the derivatives.

(Refer Slide Time: 10:45)



As you can see, this works only if the E step gives you something which you can easily maximize. In the case of Gaussian mixture, we will see that by using the complete data likelihood, we will be able to get something that we can easily maximize.

(Refer Slide Time: 11:06)



The Q function, by definition, is the expectation of the complete data likelihood under the distribution of Z given the previously guessed parameters.

$$Q\left(\vartheta, \vartheta^{(m-1)}\right) = \mathbb{E}_{Z|X,\vartheta^{(m-1)}} \log p(X, Z|\vartheta)$$

So the log likelihood here is the likelihood of x_n, z_n given the unknown parameters, over all the data points.

$$Q\left(\vartheta, \vartheta^{(m-1)}\right) = \mathbb{E}_{Z|X,\vartheta^{(m-1)}} \left[\sum_{n=1}^{N} \log p(x_n, z_n|\vartheta)\right] \qquad (2)$$

We can take this summation outside by linearity of expectations. Rewriting the the complete log likelihood, the equation assumes the form:

$$Q\left(\vartheta, \vartheta^{(m-1)}\right) = \sum_{n=1}^{N} \mathbb{E}_{Z|X,\vartheta^{(m-1)}} \left[\log \prod_{k=1}^{K} \left(\pi_k \, p(x_n|\theta_k)\right)^{\mathbb{I}(z_n=k)}\right] \qquad (3)$$

So we can derive this formally, but intuitively it is very clear. $\mathbb{I}(z_n = k)$ is just an indicator function which assumes a value of 1 when $z_n = k$, and assumes a value of 0 when $z_n \neq k$. So in this product, all the terms except one will get an exponent of 0 (and therefore, thos terms assume a value of 1). So only one term out of the K which will remain for each of the data points. The log-

likelihood comes straight away from this formula after using the indicator function here, and this gives us the complete data likelihood.

So then the product becomes a sum when we take it outside the log, and the exponent term $\mathbb{I}(z_n = k)$ comes down. Again the expectation can be brought inside by linearity, and so we get both the summations out.

$$= \sum_{n=1}^{N} \sum_{k=1}^{K} \mathbb{E}_{Z|X,\vartheta^{(m-1)}} [\mathbb{I}(z_n = k)] \log(\pi_k \, p(x_n|\theta_k)) \tag{4}$$

With respect to the distribution of Z using the previously guessed parameters $\vartheta^{(m-1)}$, $\log(\pi_k \, p(x_n|\theta_k))$ is just a constant. So the expectation is only over the indicator function $\mathbb{I}(z_n = k)$.

Expectation of an indicator function just gives us the probability, and we have the probability of $z_n = k$, again given X and the previously guessed parameter values $\vartheta^{(m-1)}$. Log just remains.

$$= \sum_{n=1}^{N} \sum_{k=1}^{K} p\left(z_n = k \middle| X, \vartheta^{(m-1)}\right) \log(\pi_k \, p(x_n|\theta_k))$$

So $p\left(z_n = k \middle| X, \vartheta^{(m-1)}\right)$ is again the responsibility, which is the posterior probability of $z_n = k$. In this case, this responsibility is not with respect to the original parameters, but this posterior probability is with respect to the guessed parameters. This has been indicated with the subscript. So it is the responsibility times the log that remains.

$$= \sum_{n=1}^{N} \sum_{k=1}^{K} \gamma(z_{nk})_{|\vartheta^{m-1}} \log(\pi_k \, p(x_n|\theta_k))$$

So, expressing the log of a product as a summation of individual log terms, we get:

$$Q(\vartheta, \vartheta^{(m-1)}) = \sum_{n=1}^{N} \sum_{k=1}^{K} \gamma(z_{nk})_{|\vartheta^{m-1}} \log \pi_k + \gamma(z_{nk})_{|\vartheta^{m-1}} \log p(x_n|\theta_k) \tag{5}$$

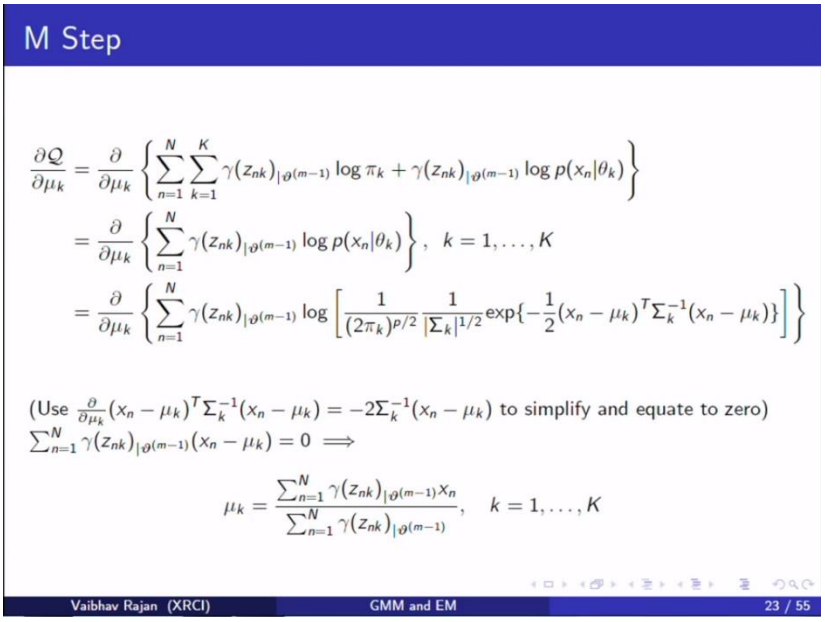
So what have we achieved? So we have got an expression for Q in terms of again the responsibility, but this time the responsibility is with respect to the guesssed parameters.

If you just look at the Q function in equation (5), you can see that it is easier to differentiate because the summations are all outside and the normal distribution term comes towards the end of the equation. The differentiation of the normal distribution term will be just like what you do in the case of fitting a single Gaussian. How did this nice mathematical form come about? It happened mainly because we were taking the complete data likelihood as shown in equation (2). The complete data likelihood gave us a nice mathematical form in equation (3), and due to the expectation, all the summations got pushed out in equation (4). So we got a nice form for Q as shown in equation (5).

So is it only because the summation is inside the logarithm, we need to do all this? Otherwise, can we can skip the whole thing ?

Yes, but that comes in many contexts, not just in the case of Gaussian mixture. In a lot of those cases, EM is useful. If we could get the maximum likelihood easily, we wouldn't need to use EM for Gaussian mixture.

(Refer Slide Time: 16:12)



Now, the M step is just that which is shown in equation (1), which is we differentiate the Q function with respect to each of the parameters. The Q function here is the same Q function which is expanded in equation (5). Now one thing to remember here is that we know the guessed parameter at the previous iteration ($\vartheta^{(m-1)}$). So we know the responsibilities. So the responsibilities are just constants in this case, and so differentiating the Q function with respect to the parameters becomes very easy.

So here we differentiate the Q function from equation (5) with respect to μ_k.

$$\frac{\partial Q}{\partial \mu_k} = \frac{\partial}{\partial \mu_k}\left\{\sum_{n=1}^{N}\sum_{k=1}^{K}\gamma(z_{nk})_{|\vartheta^{(m-1)}}\log\pi_k + \gamma(z_{nk})_{|\vartheta^{(m-1)}}\log p(x_n|\theta_k)\right\}$$

This whole first term inside the summations (which is $\boldsymbol{\gamma(z_{nk})_{|\vartheta^{(m-1)}}\log\pi_k}$) is not necessary. We focus only on the second term inside the summations (which is $\boldsymbol{\gamma(z_{nk})_{|\vartheta^{(m-1)}}\log p(x_n|\theta_k)}$).

For each of the different components, we get the entire normal distribution within the equation here.

$$\frac{\partial Q}{\partial \mu_k} = \frac{\partial}{\partial \mu_k}\left\{\sum_{n=1}^{N} \gamma(z_{nk})_{|\vartheta^{(m-1)}} \log p(x_n|\theta_k)\right\}, k = 1, \ldots, K$$

$$\frac{\partial Q}{\partial \mu_k} = \frac{\partial}{\partial \mu_k}\left\{\sum_{n=1}^{N} \gamma(z_{nk})_{|\vartheta^{(m-1)}} \log\left[\frac{1}{(2\boldsymbol{\pi})^{\frac{p}{2}}} \frac{1}{|\Sigma_k|^{\frac{1}{2}}} \exp\left\{-\frac{1}{2}\,(x_n - \mu_k)^T\, \Sigma_k^{-1}\,(x_n - \mu_k)\right\}\right]\right\}$$

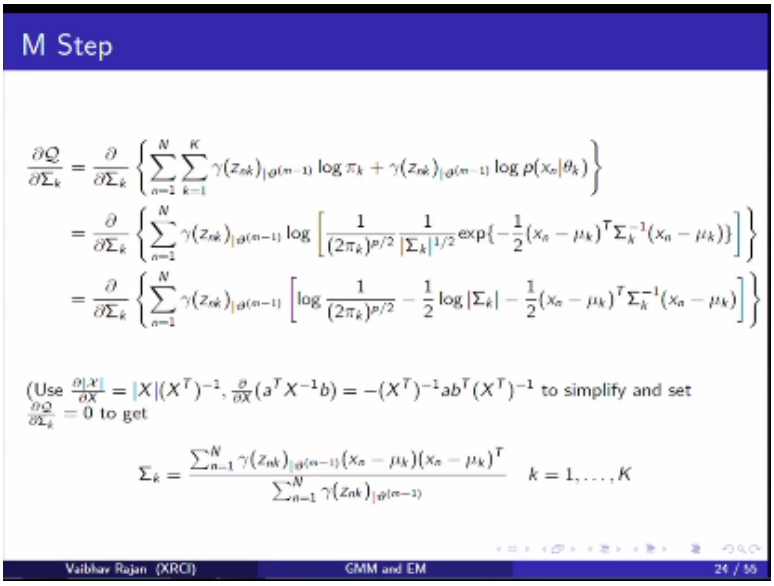
There are no summations inside the logarithm. This is exactly the same derivation as for a single Gaussian. We use matrix derivatives to get very simple forms here.

Substituting $\frac{\partial Q}{\partial \mu_k} = 0$, we get:

$$\mu_k = \frac{\sum_{n=1}^{N} \gamma(z_{nk})_{|\vartheta^{(m-1)}}\, x_n}{\sum_{n=1}^{N} \gamma(z_{nk})_{|\vartheta^{(m-1)}}}, k = 1, \ldots, K$$

We will see that we again get the same form for μ_k that we saw earlier for our adhoc iterative algorithm except that these responsibilities are with respect to the previously guessed parameters.

(Refer Slide Time: 17:43)



Here, similar to what we did for μ_k, we find the derivative of Q from equation (5) with respect to Σ_k and equate it to zero.

$$\frac{\partial Q}{\partial \Sigma_k} = \frac{\partial}{\partial \Sigma_k}\left\{\sum_{n=1}^{N}\sum_{k=1}^{K} \gamma(z_{nk})_{|\vartheta^{(m-1)}} \log \pi_k + \gamma(z_{nk})_{|\vartheta^{(m-1)}} \log p(x_n|\theta_k)\right\}$$

Again, we do not need to worry about the first term within the summations. We only find the derivative for the second term within the summations.

$$\frac{\partial Q}{\partial \Sigma_k} = \frac{\partial}{\partial \Sigma_k}\left\{\gamma(z_{nk})_{|\vartheta^{(m-1)}} \log\left[\frac{1}{(2\boldsymbol{\pi})^{\frac{p}{2}}}\frac{1}{|\Sigma_k|^{\frac{1}{2}}}\exp\left\{-\frac{1}{2}\,(x_n-\mu_k)^T\,\Sigma_k^{-1}\,(x_n-\mu_k)\right\}\right]\right\}$$

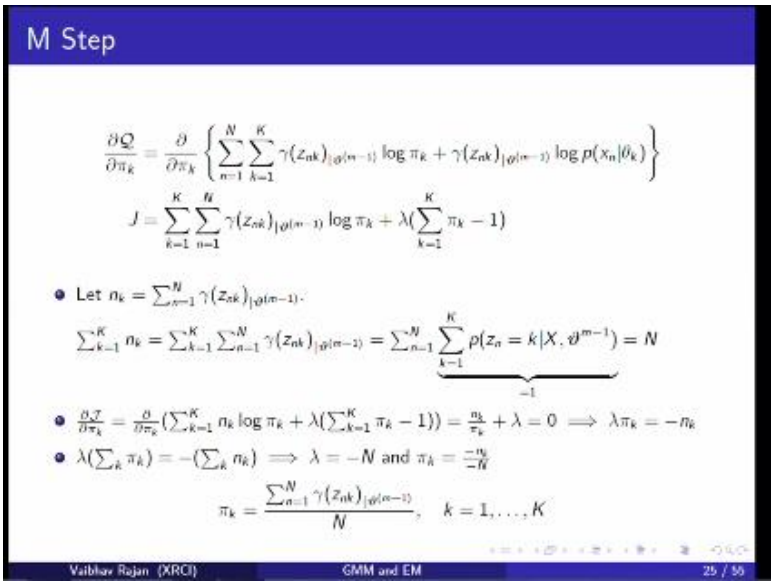
You can simplify this further by first applying the logarithm here for each of these parts.

$$\frac{\partial Q}{\partial \Sigma_k} = \frac{\partial}{\partial \Sigma_k}\left\{\sum_{n=1}^{N} \gamma(z_{nk})_{|\vartheta^{(m-1)}}\left[\log\frac{1}{(2\boldsymbol{\pi})^{\frac{p}{2}}} - \frac{1}{2}\log|\Sigma_k| - \frac{1}{2}\,(x_n-\mu_k)^T\,\Sigma_k^{-1}(x_n-\mu_k)\right]\right\}$$

Then, the derivative for the determinant is given by a simple formula. We can apply the derivative formula here, and by setting $\frac{\partial Q}{\partial \Sigma_k} = 0$, we get back the same form for Σ_k which we found earlier.

$$\Sigma_k = \frac{\sum_{n=1}^{N} \gamma(z_{nk})_{|\vartheta^{(m-1)}}\,(x_n-\mu_k)(x_n-\mu_k)^T}{\sum_{n=1}^{N} \gamma(z_{nk})_{|\vartheta^{(m-1)}}}, k = 1, \ldots, K$$

(Refer Slide Time: 18:22)



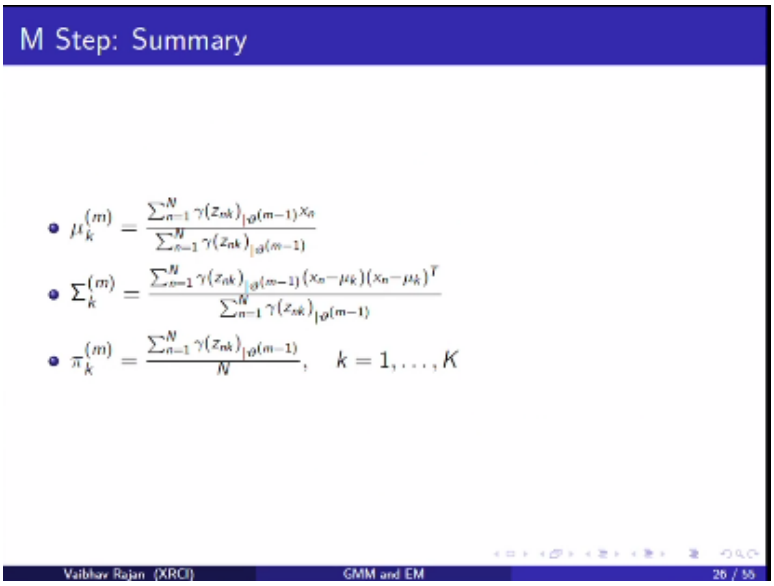
So similarly we perform the computations for the M step for π_k. However, this time in $\frac{\partial Q}{\partial \pi_k}$ the second term within the summations (which is $\boldsymbol{\gamma(z_{nk})_{|\vartheta^{(m-1)}}\log p(x_n|\theta_k)}$) is a constant with respect to π_k and therefore goes away. We focus on the differentitation of the first term within the summations (which is $\boldsymbol{\gamma(z_{nk})_{|\vartheta^{(m-1)}}\log \pi_k}$). Since, the π_ks must satisfy the constraint $\sum_{k=1}^{K}\pi_k = 1$, we use Lagrange multipliers to get the formulation for π_k. The langrange function J can be written as:

$$J = \sum_{k=1}^{K}\sum_{n=1}^{N}\gamma(z_{nk})_{|\vartheta^{(m-1)}}\log\pi_k + \lambda\left(\sum_{k=1}^{K}\pi_k - 1\right)$$

By differentiating the langrage function J with respect to π_k, and setting the derivative to 0, we get:

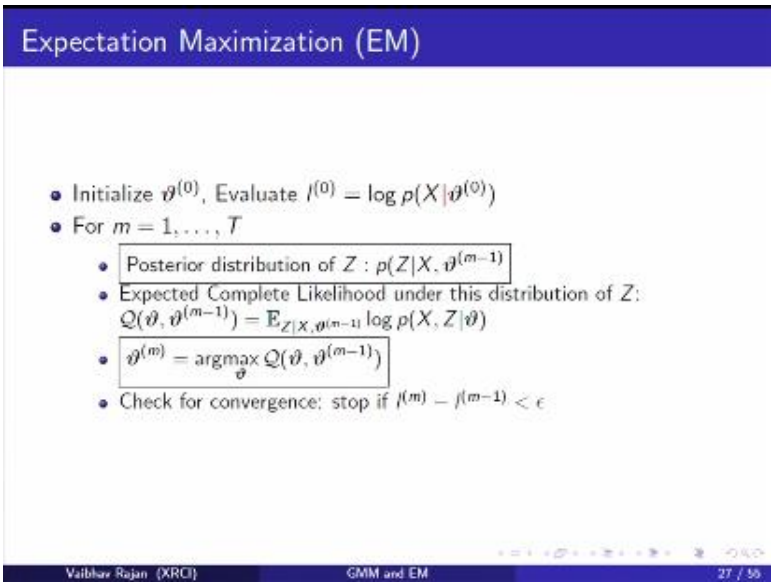
$$\pi_k = \frac{\sum_{n=1}^{N}\gamma(z_{nk})_{|\vartheta^{(m-1)}}}{N}, k = 1, \ldots, K$$

(Refer Slide Time: 18:50)



In the previous lecture, we first found sformulas for μ_k, Σ_k and π_k by assuming that we know the responsibilities. Then, in this lecture, we used the E and M steps to find exactly the same formulas for μ_k, Σ_k and π_k that we had found earlier in the previous lecture.

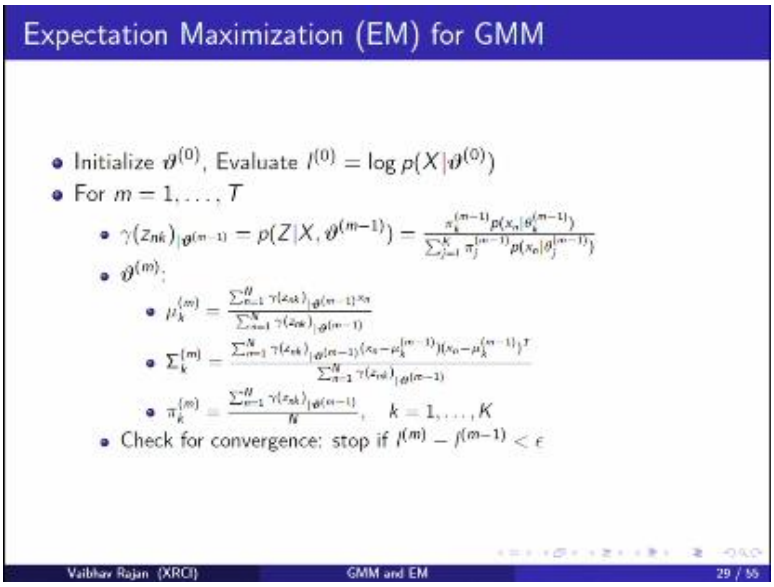
(Refer Slide Time: 19:13)



We plug the formulas for μ_k, Σ_k and π_k into the EM framework. In the EM framework, we start with a guess for these parameters. We then iterate by first finding the posterior distribution of Z

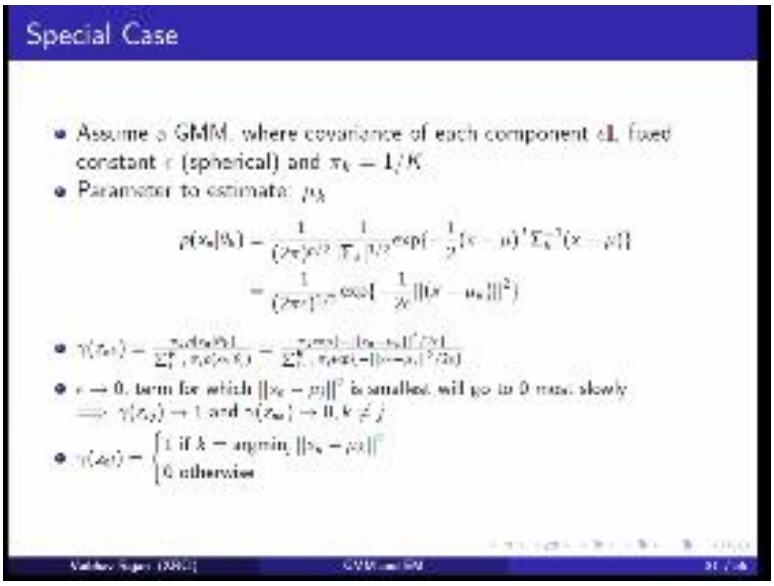
(which gives us the responsibilities and can be represented as $p(Z|X, \vartheta^{(m-1)})$), and then we find the expected complete likelihood under this distribution of Z (which can be represented as $Q(\vartheta, \vartheta^{(m-1)}) = \mathbb{E}_{Z|X,\vartheta^{(m-1)}} \log p(X, Z|\vartheta)$), and we finally maximize the Q function to get the new guesses for the parameters.

(Refer Slide Time: 19:43)



This gives us the next set of guesses, and this iteratively we can, we can check for convergence. When the likelihood does not change much we stop, and that is the EM algorithm for GMM. I have still not told you why this works, but we will see that, we will see the theoretical properties of why this is, why this works well. Yeah. I just wanted to show you that what we have got through, by doing all the math for EM is exactly the same as what we found during the iterative algorithm that we guessed.

(Refer Slide Time: 20:31)



Let's look at a special case. Let's assume a Gaussian mixture model where the covariance of each component is fixed to be $\epsilon\mathbb{I}$, where ϵ is a fixed constant and $\mathbb{I}$ is the identity matrix. Covariance matrix of $\Sigma_k = \epsilon\mathbb{I}$ will give us a spherical Gaussian. We also fixed each of the π_k to be $\frac{1}{K}$. So each component gives exactly the same contribution towards the Gaussian mixture. Now the only parameter to estimate is the μ_ks.

Since $\Sigma_k = \epsilon\mathbb{I}$, the formula for the normal distribution simplifies to:

$$p(x|\theta_k) = \frac{1}{(2\boldsymbol{\pi})^{\frac{p}{2}}} \frac{1}{|\Sigma_k|^{\frac{1}{2}}} \exp\left\{-\frac{1}{2}\ (x-\mu_k)^T\ \Sigma_k^{-1}\ (x-\mu_k)\ )\right\}$$

$$p(x|\theta_k) = \frac{1}{(2\boldsymbol{\pi}\epsilon)^{\frac{p}{2}}} \exp\left\{-\frac{1}{2\epsilon}\ \|(x-\mu_k)\|^2\right\} \qquad (6)$$

The formula for the responsibility also simplifies. We plug $p(x_n|\theta_k)$ from equation (6) and also substitute $\pi_k = \frac{1}{K}$ into the formula for responsibilities $\gamma(z_{nk})$, and we get:

$$\gamma(z_{nk}) = \frac{\pi_k\ p(x_n|\theta_k)}{\sum_{j=1}^{K} \pi_j\ p(x_n|\theta_j)} = \frac{\exp\{-\|x_n-\mu_k\|^2/2\epsilon\ \}}{\sum_{j=1}^{K} \exp\left\{-\|x_n-\mu_j\|^2/2\epsilon\right\}} \qquad (7)$$

Now if we look at this expression in equation (7), and see what happens to the denominator as $\epsilon \to 0$, as $\epsilon \to 0$, the term for which the difference $\|x_n - \mu_j\|^2$ is smallest will go to 0 most slowly. So the responsibility for x_n will tend to 1 for that particular component (which is the j^{th} component in this case) because the numerator will be equal to the denominator in the limits, and for all other components (for which $k \neq j$), the responsibility will go to 0.

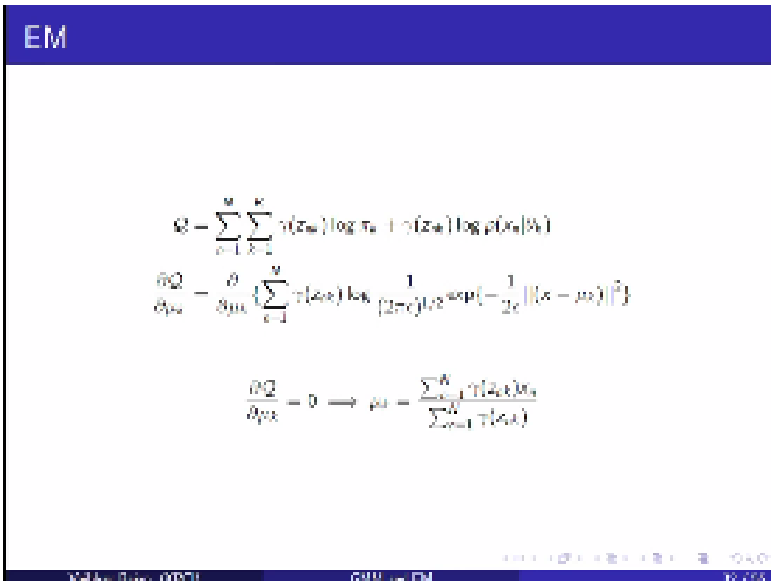
$$\gamma(z_{nj}) \to 1 \quad \text{and} \quad \gamma(z_{nk}) \to 0, \ \forall\, k \neq j$$

So this is the special case of hard clustering that was mentioned earlier. What it turns out to be is just setting the responsibility of x_n to 1 for that component for which the data point is closest to the mean, and setting the responsibility of x_n to zero for all other components.

$$\gamma(z_{nk}) = \begin{cases} 1 \; if \; k = argmin_j \; \|x_n - \mu_j\|^2 \\ 0 \; otherwise \end{cases}$$

The responsibility is just the posterior probability of z_n being equal to k. For the data point x_n, we want to know which component it has come from. For the data point x_n, the responsibility is one for that component whose mean the data point is closest to, and it is 0 for all others components. This means that if you look at the data and look at x_n, and want to know the posterior probability of which component it came from, it is that component whose mean that data point is closest to.

(Refer Slide Time: 24:08)



Let us do the EM for this special case of hard clustering in which for each of the components, we have set $\Sigma_k = \epsilon\mathbb{I}$ and $\pi_k = 1/K$. The first step in EM is to calculate Q.

From equation (5), we have:

$$Q = \sum_{n=1}^{N}\sum_{k=1}^{K} \gamma(z_{nk}) \log \pi_k + \gamma(z_{nk}) \log p(x_n|\theta_k)$$

The formula for Q is the same as before because we are doing Gaussian mixture. It's just a special Gaussian mixture.

Substituting $\pi_k = \frac{1}{K}$, and replacing $\log p(x_n|\theta_k)$ with the expansion from equation (6) for the this special case of Gaussian, we have:

$$Q = \sum_{n=1}^{N}\sum_{k=1}^{K} \gamma(z_{nk}) \log \frac{1}{K} + \gamma(z_{nk}) \log \frac{1}{(2\boldsymbol{\pi}\epsilon)^{\frac{p}{2}}} \exp\left\{-\frac{1}{2\epsilon} \|(x_n - \mu_k)\|^2\right\}$$

We do the differentiation of the Q function which has this simplified normal distribution in it.

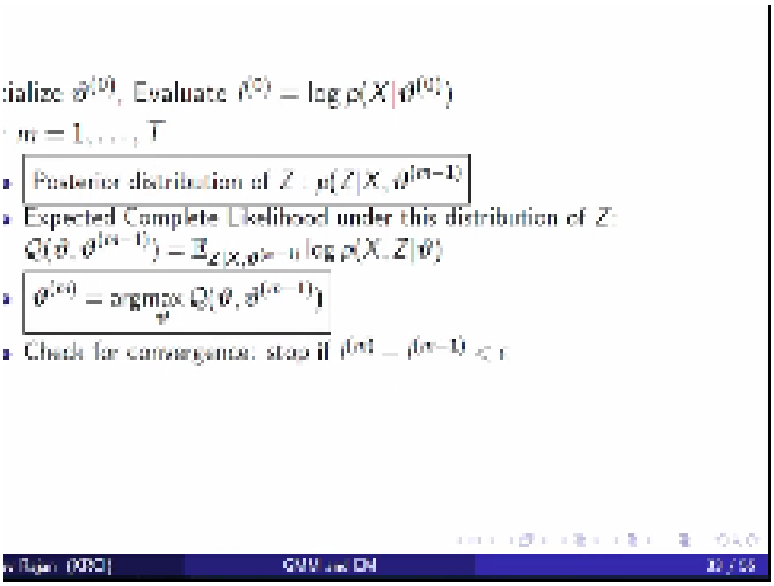
$$\frac{\partial Q}{\partial \mu_k} = \frac{\partial}{\partial \mu_k}\left\{\sum_{n=1}^{N} \gamma(z_{nk}) \log \frac{1}{(2\boldsymbol{\pi}\epsilon)^{\frac{p}{2}}} \exp\left\{-\frac{1}{2\epsilon}\ \|(x_n - \mu_k)\|^2\right\}\right\}$$

$$\frac{\partial Q}{\partial \mu_k} = 0\ \Rightarrow \mu_k = \frac{\sum_{n=1}^{N} \gamma(z_{nk})\ x_n}{\sum_{n=1}^{N} \gamma(z_{nk})}$$

We again get the same formula for μ_k, but the only difference is that this responsibility is defined in the way it was described earlier (for hard clustetring).

What is it basically saying? So for the k th component, only the responsibilities of the data points assigned to the k^{th}component will be 1, and responsibilities of the same data points corresponding to other components will be 0. So we take all the data points that are assigned to the k^{th} component, take the mean of those data points, and update the mean parameter of k^{th} component, μ_k, with this computed mean value.

(Refer Slide Time: 25:13)



This is the general EM algorithm. We first calculate the posterior distribution of Z, which we saw is exactly given by:

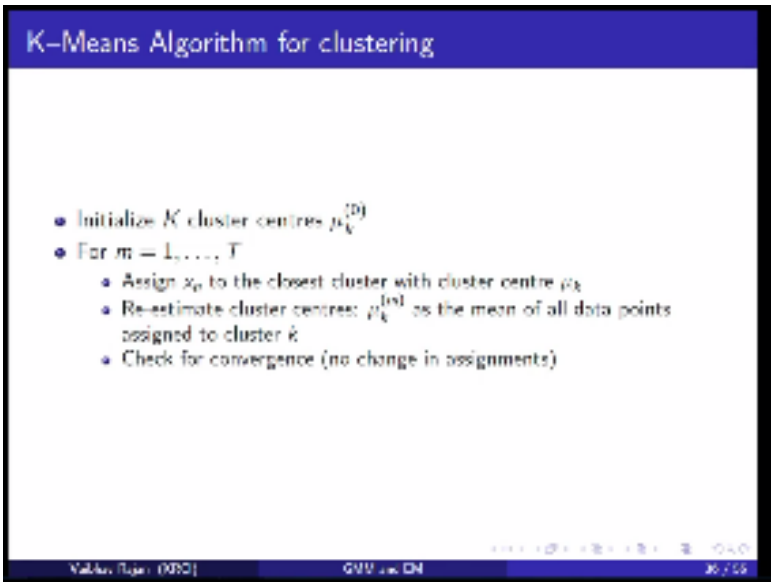
$$\gamma(z_{nk}) = \begin{cases} 1 \; if \; k = argmin_j \left\| x_n - \mu_j \right\|^2 \\ 0 \; otherwise \end{cases}$$

We assign the latent variable of x_n to the closest mean, and then set the new mean as the mean of all data points with the same latent variable. This is exactly also the k-means algorithm.

So we are assigning x_n to the closest cluster, with the cluster center μ_k, and then reestimating the cluster centers as the mean of all data points that are assigned to that cluster. So k-means is just a special case of Gaussian mixture, where the covariance matrix is epsilon times the identity matrix. That is why we just have to compute the means, and do not have to worry about the covariance.

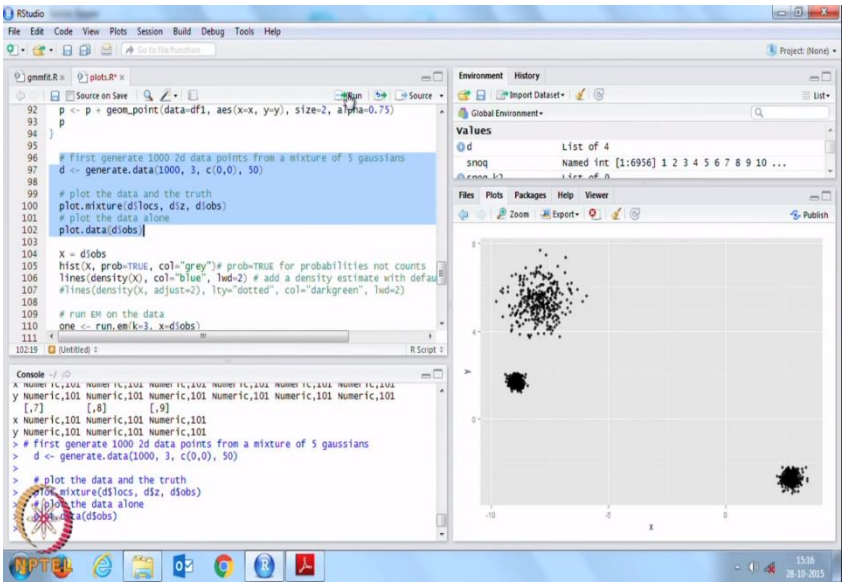
This entire procedure follows the same framework of EM that we saw.

(Refer Slide Time: 26:31)



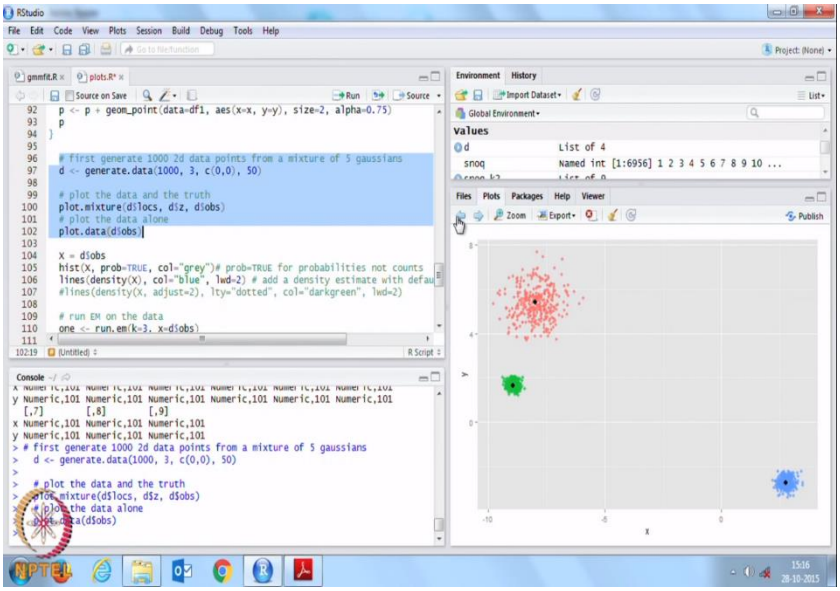
So I think we will stop here because after this we will talk about all the theoretical properties of EM. I wanted to show you one more thing.

(Refer Slide Time: 26:59)



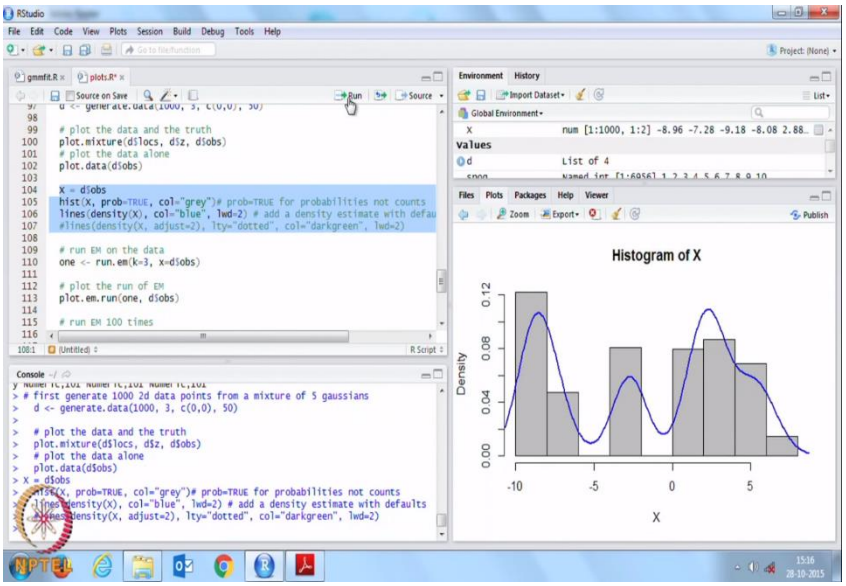
Let us consider this data.

(Refer Slide Time: 27:05)



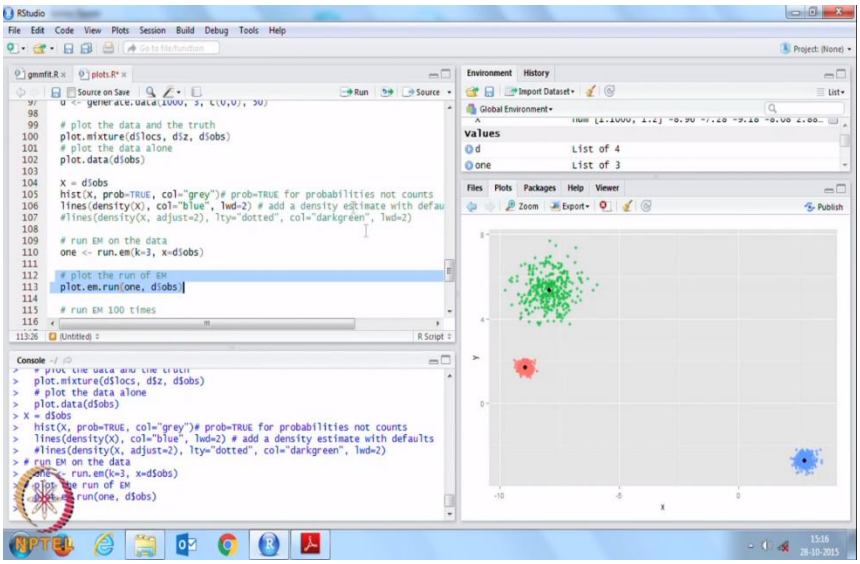
The data is generated from three Gaussians. The three Gaussians and their cluster centers are shown here. The previous figure shows how the data looks.

(Refer Slide Time: 27:18)



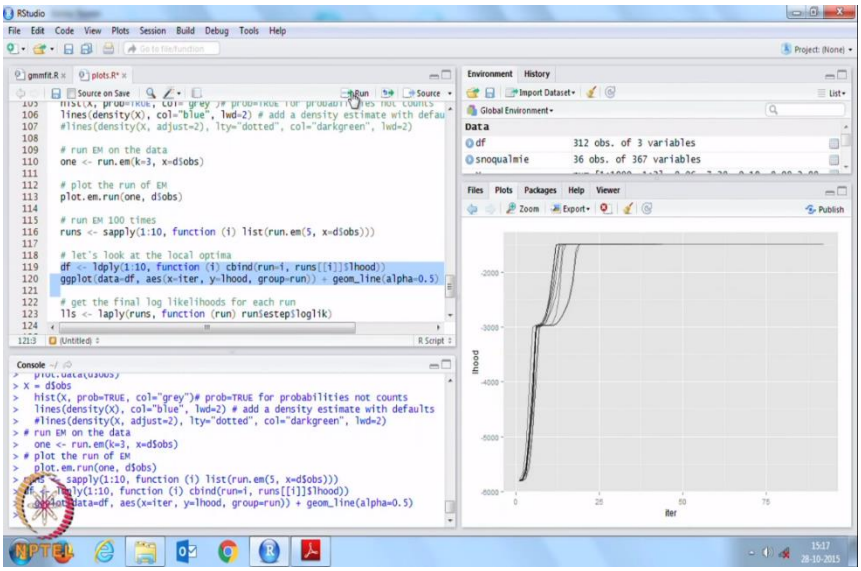
This is the density plot corresponding to the X coordinates of the data.

(Refer Slide Time: 27:30)



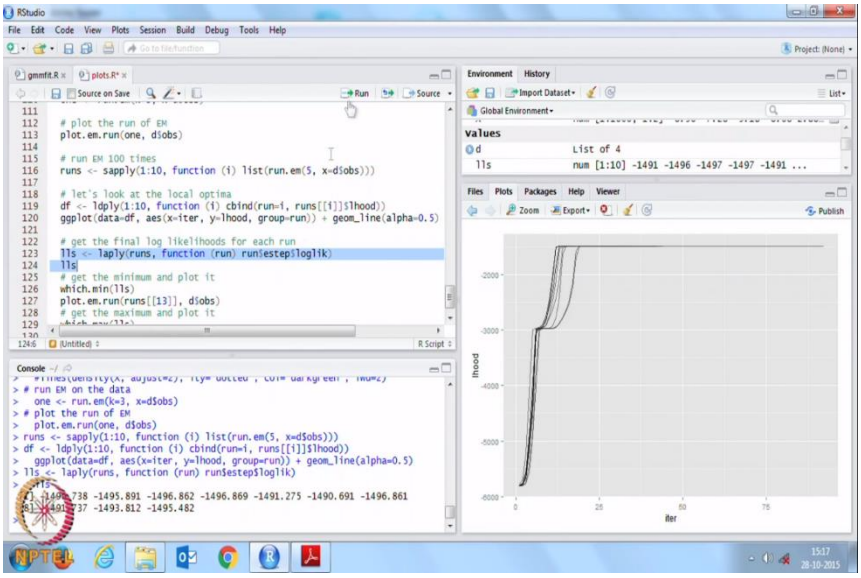
If we run EM once on it, and plot it, we see that it recovers the means and covariances exactly the way the data was generated.

(Refer Slide Time: 28:02)



Now let's run the EM algorithm ten times, and plot it. In this plot, on the x axis we have iterations and on the y axis we have the likelihood. In each iteration, the likelihood keeps increasing until it reaches a point and remains steady there. So this is a property of the EM algorithm which we will prove in the coming lecture.

(Refer Slide Time: 28:35)



Let us look at these ten likelihood values. The 10 negative log-likelihood values corresponding to when we stopped the iteration for each of the 10 runs are:

-1490.738	-1495.891	-1496.862	-1496.869	-1491.275
-1490.691	-1496.861	-1491.737	-1493.812	-1495.482

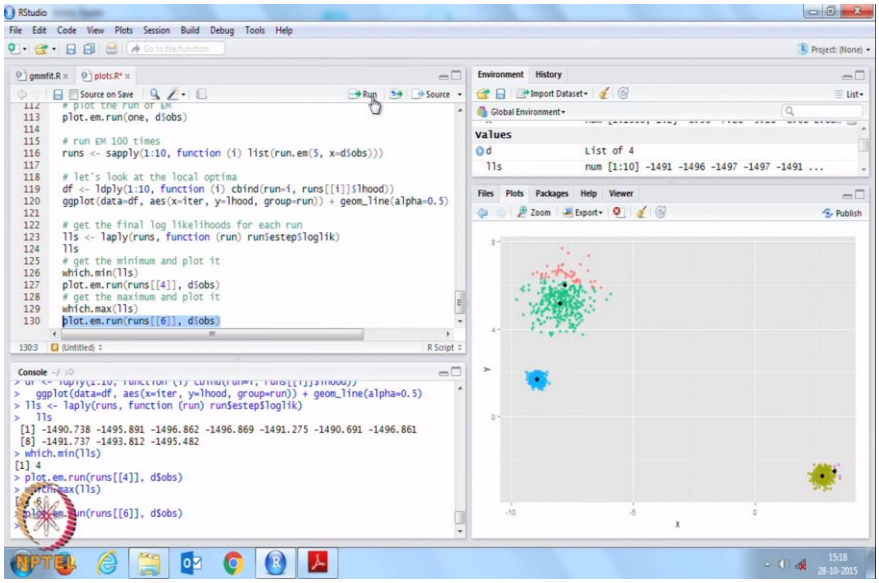
(Refer Slide Time: 28:54)

We always have a hard bound T on the number of iterations because sometimes the likelihood may not converge. So we usually give an upper bound on the number of iterations as well and stop it there.

When we ran EM ten times on the data, the 10 final negative log-likelihood values were -1490.738, -1495.891, -1496.862, -**1496.869**, -1491.275, -1490.691, -1496.861, -1491.737, -1493.812 and -1495.482. The minimum negative log likelihood was achieved for the fourth run of the EM.

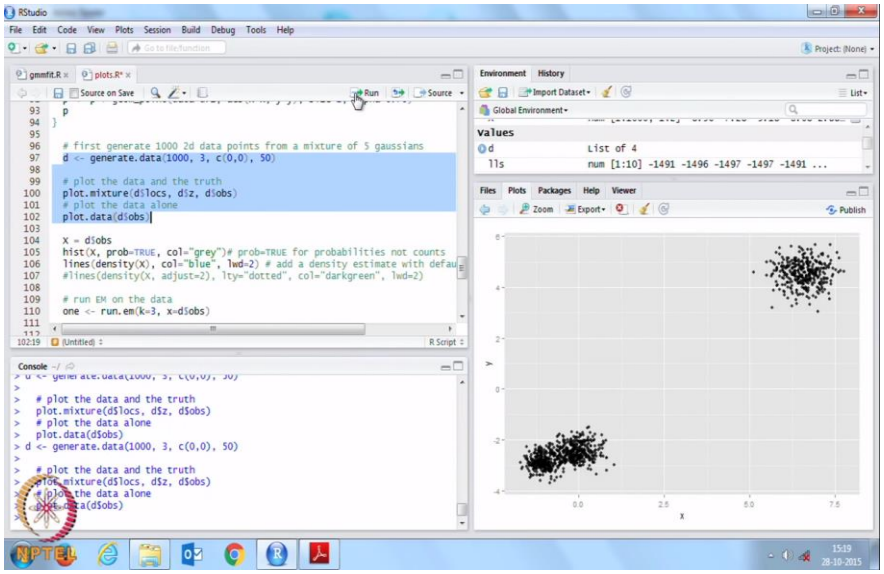
This is a very easy case.

(Refer Slide Time: 29:43)



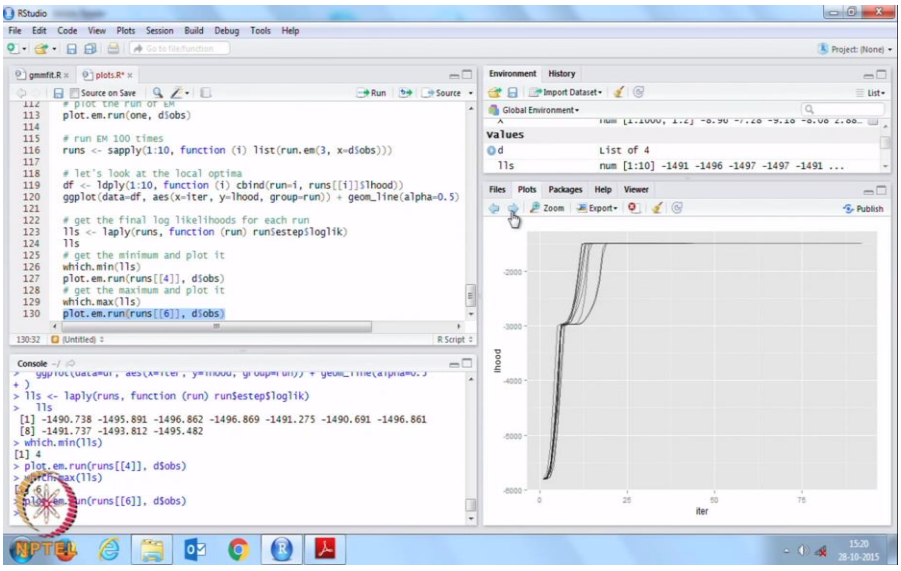
I gave $k = 5$, so it has estimated five different components. The means of the components are shown here. So it has tried to fit five Gaussians.

(Refer Slide Time: 30:35)



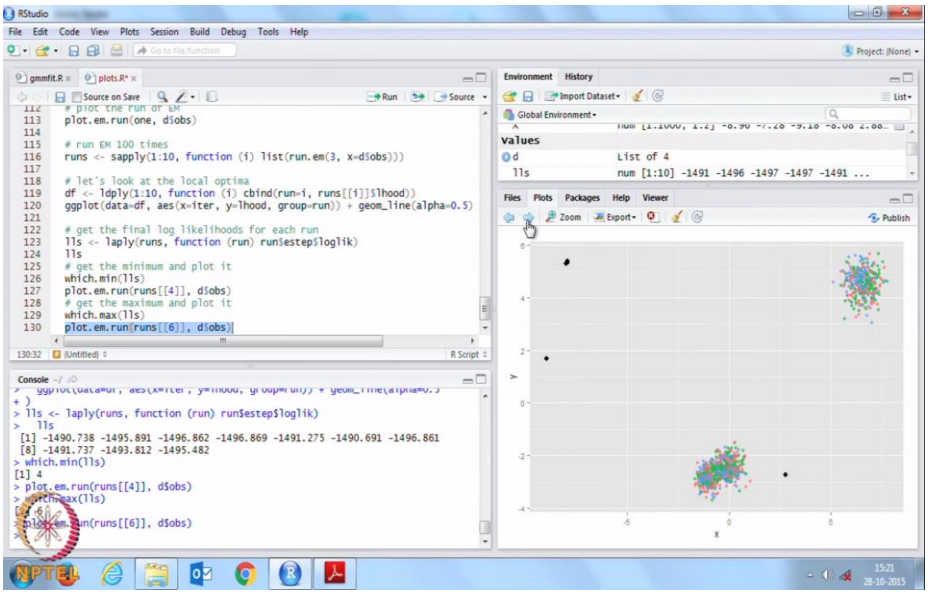
So this is a more difficult case because the three components are not very well separated. Now we run the EM algorithm ten times on this data. In this case, the data is as shown here.

(Refer Slide Time: 31:45)



EM ran, the likelihood always increased.

(Refer Slide Time: 31:48)



But what it estimated is shown here. So when the data is very well separated, like we saw earlier, EM will usually give very good results, but when the data is not very well separated like this, it starts giving very weird results. But no matter what it does, the likelihoods will always increase.

IIT Madras production

Funded by

Department of the higher education

Ministry of the human resource department

Government of India

www.nptel.ac.in

NPTEL

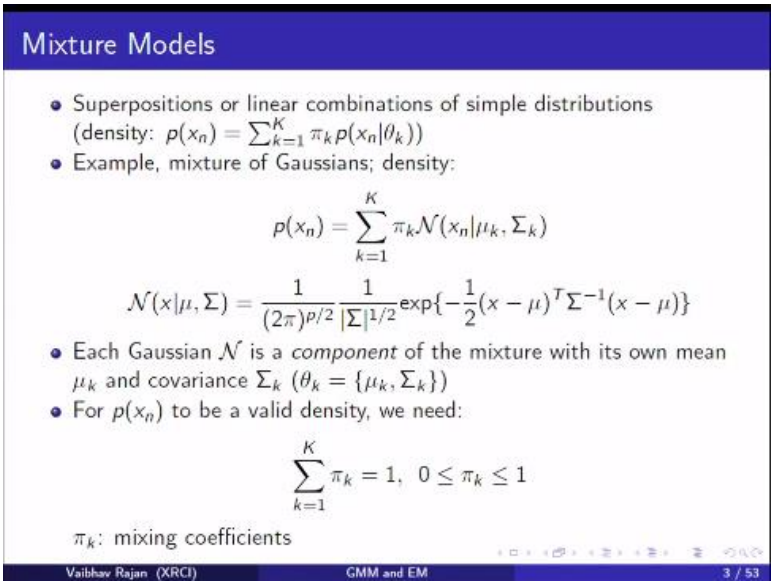
NPTEL ONLINE CERTIFICATION COURSE

Introduction to Machine Learning

Lecture-78
Expectation Maximization Continued

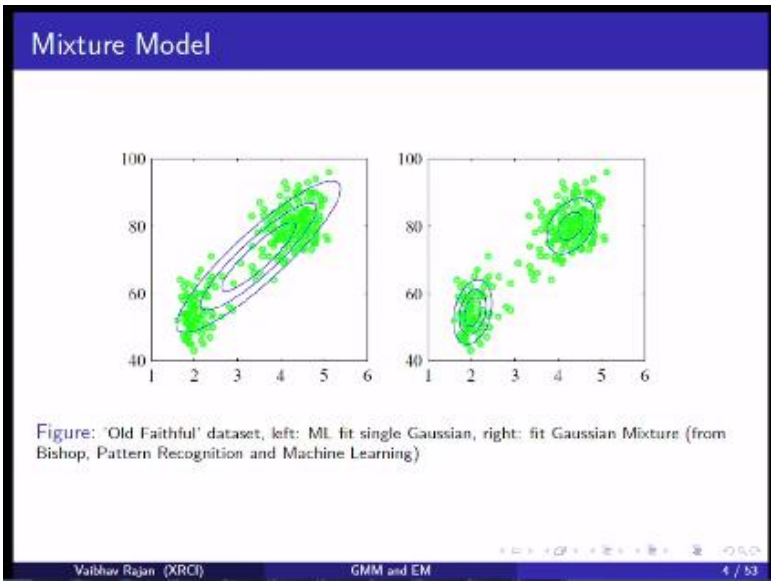
Prof: Balaraman Ravindran
Computer Science and Engineering
Indian Institute of Technology Madras

(Refer Slide Time: 00:19)



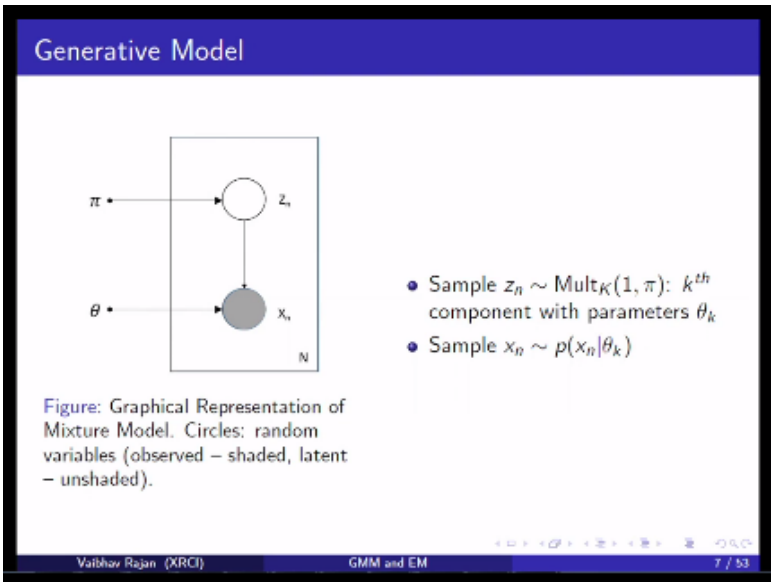
So we started with defining Gaussian mixture models which are just superposition of K different Gaussians. However in a general mixture model, we can use any other probability distribution instead of the Gaussian. The three important set of parameters in a GMM are the mixture weights π_ks, the mean matrices $\mu_k s$ and the covariance matrices Σ_ks of each of the K Gaussians (or components). In this lecture, we will also see how to estimate the value of K.

(Refer Slide Time: 00:56)



We saw some examples of how to fit Gaussian mixture models. We saw that Gaussian mixture models are naturally good models when there is a cluster structure in the data because then each of those clusters can be nicely fitted with a Gaussian.

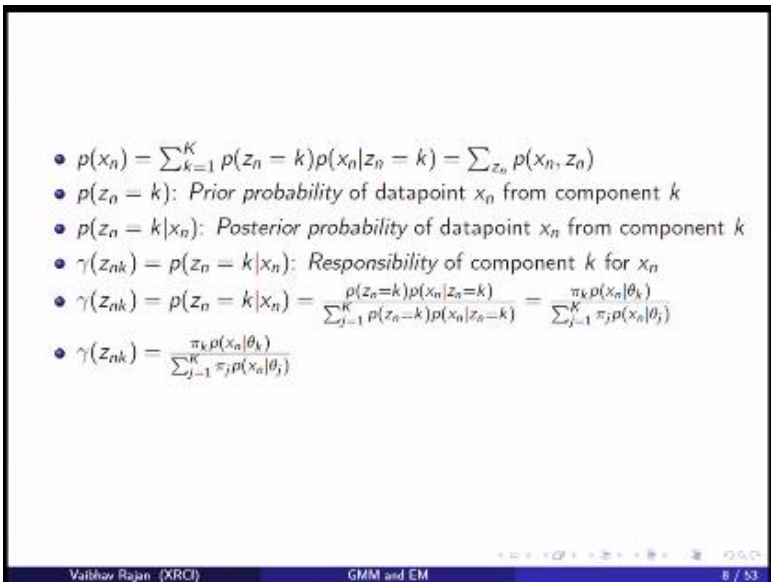
(Refer Slide Time: 01:15)



We saw that the Gaussian mixture model can be very intuitively explained through the generative procedure. Here we assume that there is a latent variable z_n that basically tells us which Gaussian

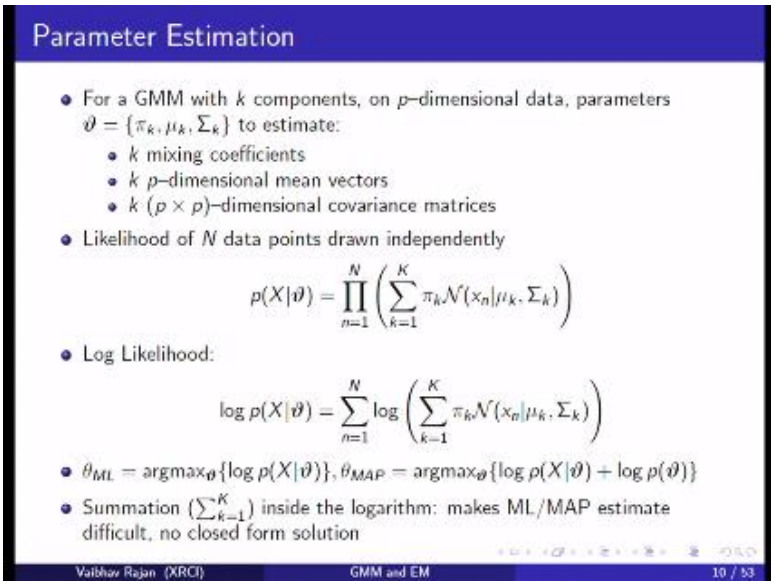
to pick. Then once we pick that Gaussian, we generate (or sample) our data point x_n from that particular Gaussian. This generative procedure is very important to remember as it makes sense of a lot of the math.

(Refer Slide Time: 01:50)



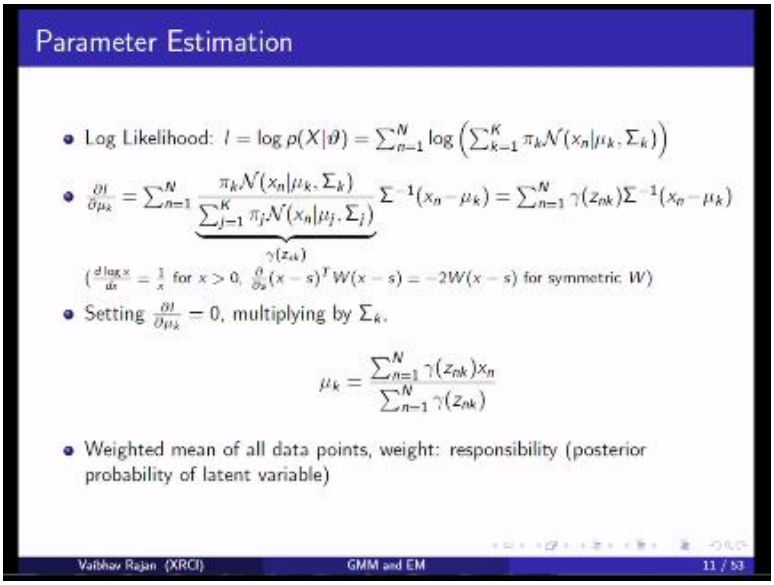
Then we saw the posterior probability of the lantent variable z_n taking value k for our datapoint x_n, which is also called the responsibility of component k for x_n. We saw that it was repeatedly coming up in all our calculations.

(Refer Slide Time: 02:11)



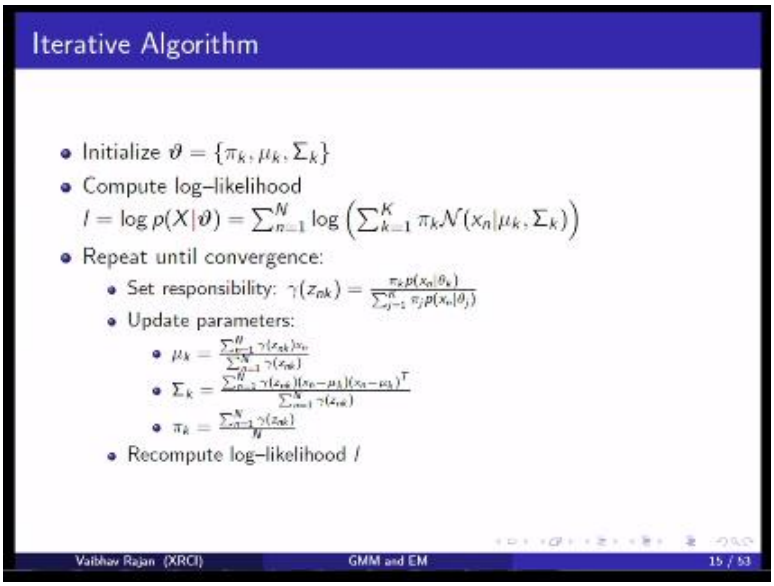
Assuming that we know the number of components in the data (which is given by K), we need to estimate the parameters π_k, μ_k and Σ_k for each of the Gaussians.

(Refer Slide Time: 02:23)



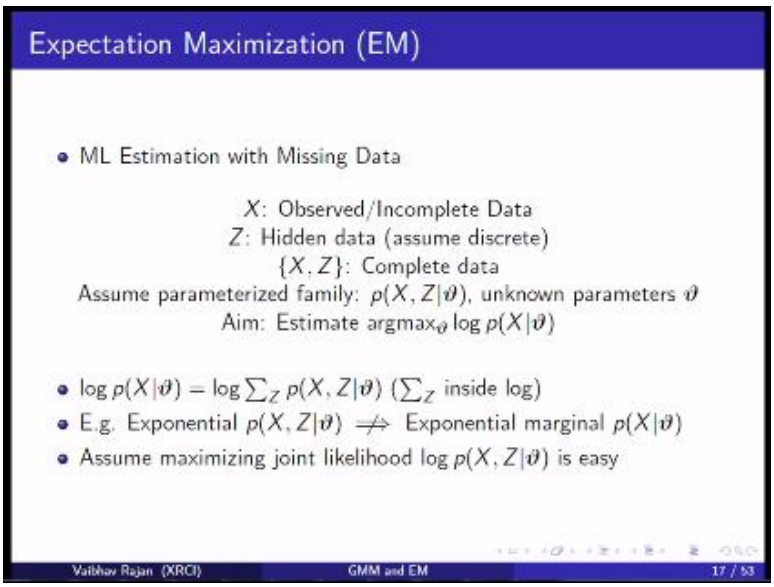
We initially saw that if we assume that we know the responsibilities $\gamma(z_{nk})$, then the math works out very nicely, and we get very intuitive forms for the different parameters π_k, μ_k and Σ_k.

(Refer Slide Time: 02:49)



We then designed an iterative algorithm that essentially first guesses the parameters $\vartheta = \{\pi_k, \mu_k, \Sigma_k\}$, then computes the responsibilities $\gamma(z_{nk})$ using these guessed parameters, and then refines the guesses for the parameters in each iteration. We later saw that this iterative algorithm is actually the EM algorithm for Gaussian mixture models.

(Refer Slide Time: 03:09)

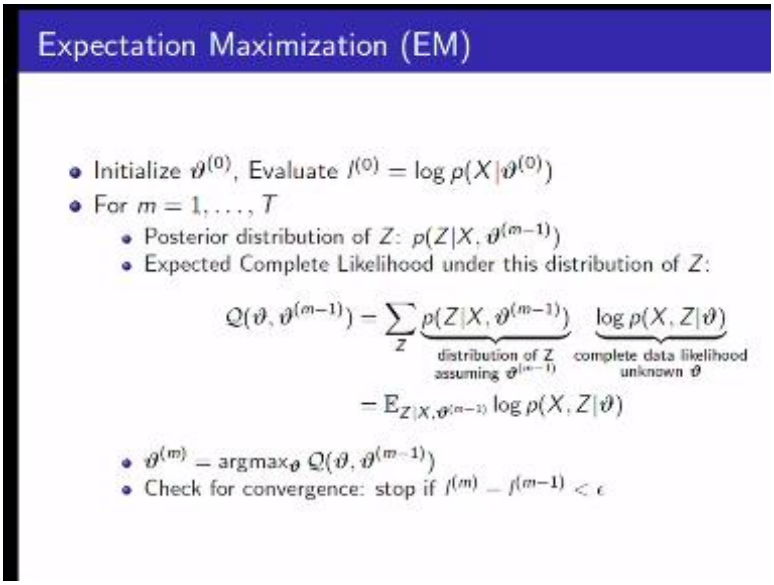


In general EM had been proposed for data which had some hidden datapoints which are not known when we get the data set. We denote that hidden data by Z. For the purpose of this discussion, we have assumed that Z is discrete.

Later we saw that in the case of the Gaussian mixture models we can take latent variables to be hidden. This is a common trick used in many other models.

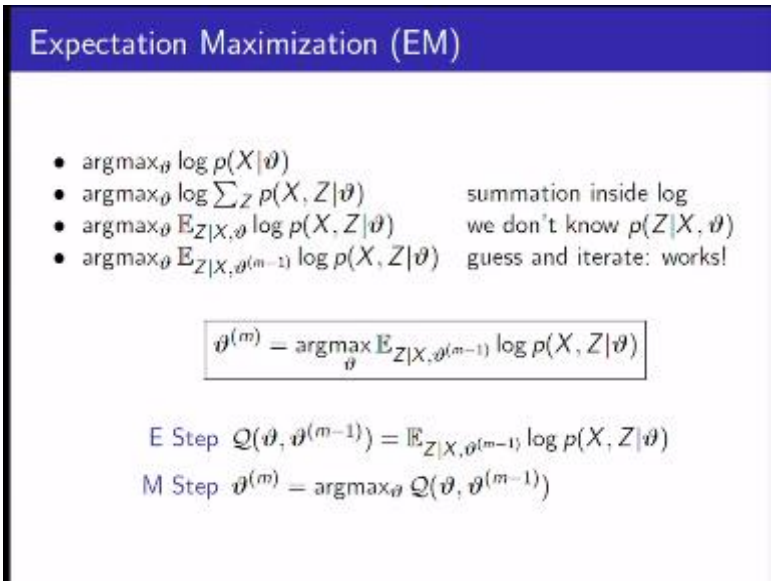
We then saw that EM is a good approach to take when the the complete data likelihood (or the joint likelihood) can be easily parameterized. If we make this assumption then we see that we can get the marginal likelihood also.

(Refer Slide Time: 04:02)



The key idea in EM is that we take the expectation of the log likelihood of the complete data under the distribution of latent variables assuming the guesses of the parameters that we had made.

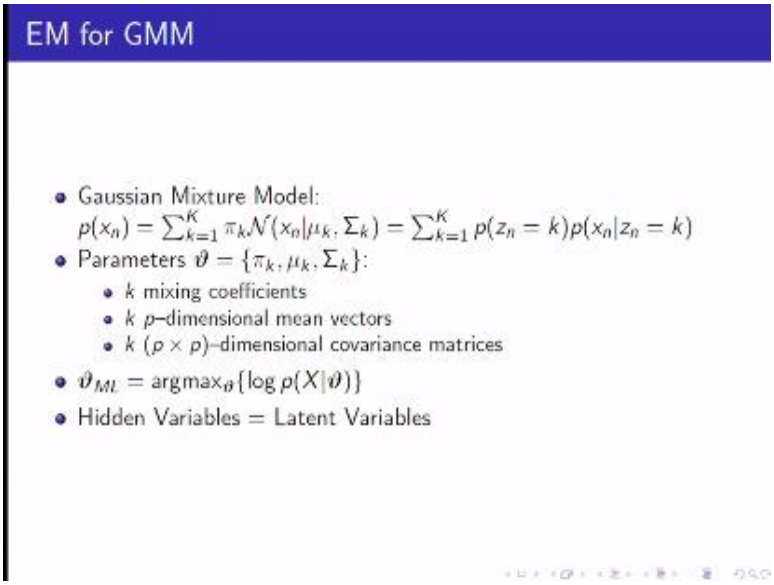
(Refer Slide Time: 04:27)



Instead of computing the maximum likelihood, we compute the parameters that maximizes the expectation $\mathbb{E}_{Z|X,\vartheta^{(m-1)}} \log p(X, Z|\vartheta)$. This is the key idea of EM. The main take away from this class should be the formula given by:

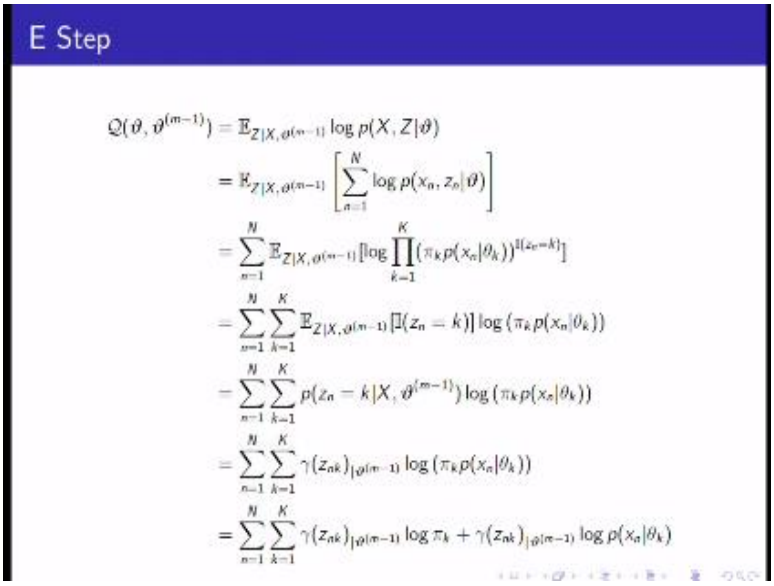
$$\vartheta^{(m)} = \operatorname*{argmin}_\vartheta \mathbb{E}_{Z|X,\vartheta^{(m-1)}} \log p(X, Z|\vartheta)$$

(Refer Slide Time: 04:49)



So then we saw that if we use this formulation, then for Gaussian mixture models, we essentially get back the iterative algorithm that we had guessed.

(Refer Slide Time: 05:04)

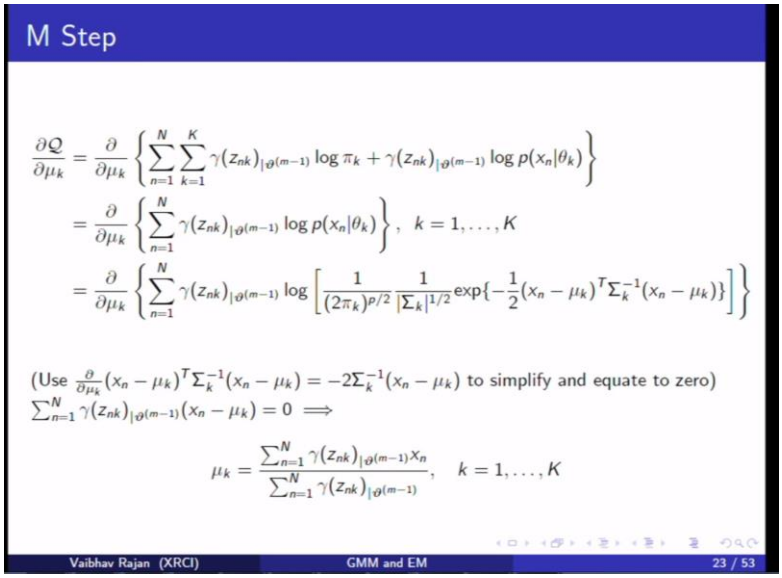


The expectation $\mathbb{E}_{Z|X,\vartheta^{(m-1)}} \log p(X, Z|\vartheta)$ is also called Q function in the literature. We get a very nice form for the Q function because of two reasons:

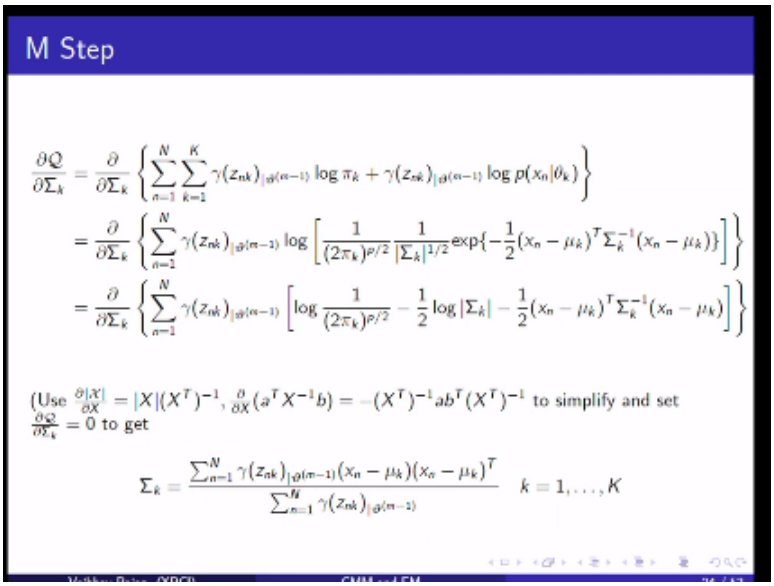
i. we are using an expectation operator which pushes the summation to the outside,
ii. we get the logarithm of the Gaussian without any summation inside because the expectation pushes it outside.

That were the reasons why the math worked out.

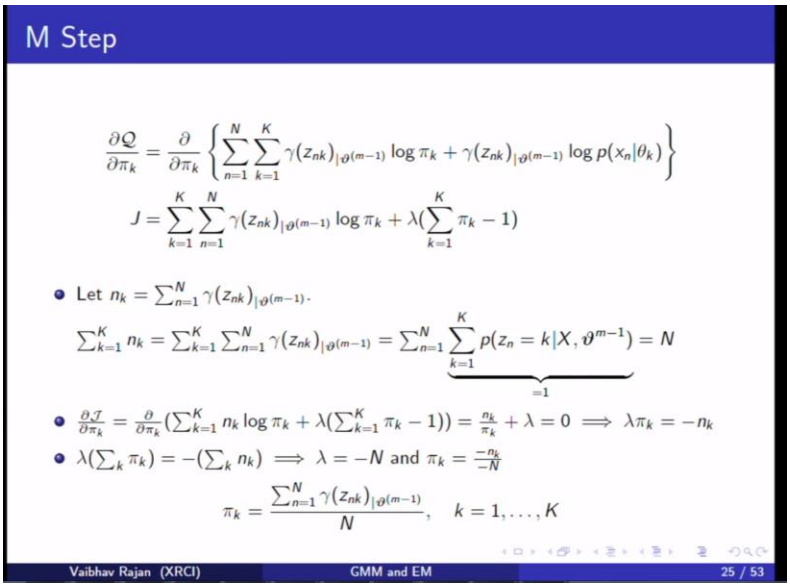
(Refer Slide Time: 05: 37)



(Refer Slide Time: 05: 41)

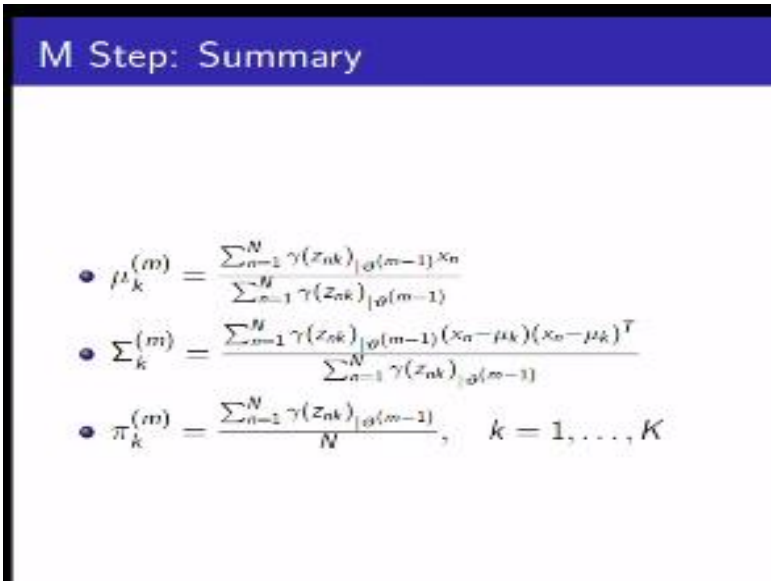


(Refer Slide Time: 05: 44)



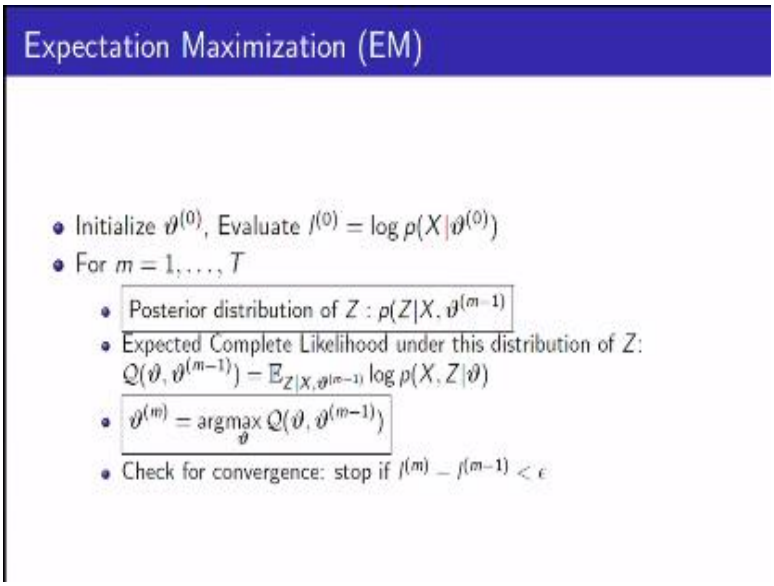
The derivatives of the Q function with respect to each of the parameters μ_k, Σ_k, and π_k became very easy to calculate for the case of Gaussian.

(Refer Slide Time: 05:46)



We essentially got back the same formulas for μ_k, Σ_k and π_k that we had guessed earlier, asumming that we know the responsibilities.

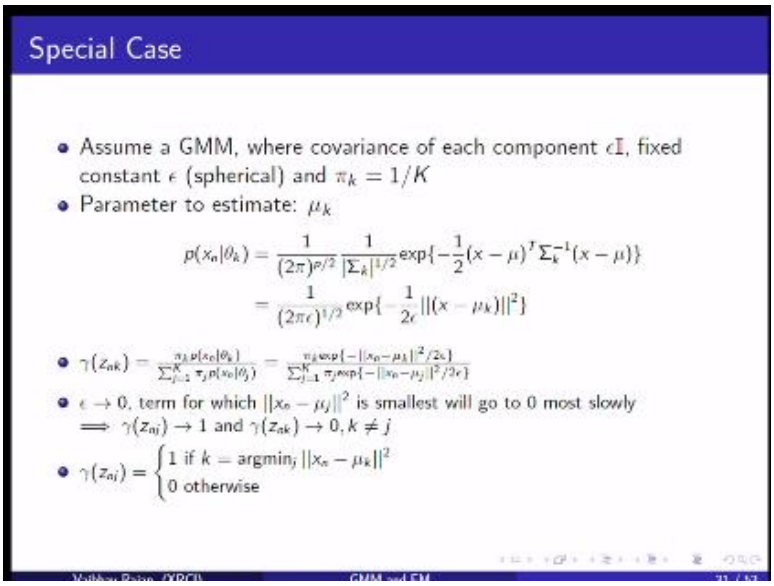
(Refer Slide Time: 06:00)



So the general EM algorithm is this - guess the posterior distribution of the hidden data (or the latent variables) Z, and then refine your guess by maximizing the Q function, which is the expectation of the complete data likelihood under the distribution of Z with your current guess.

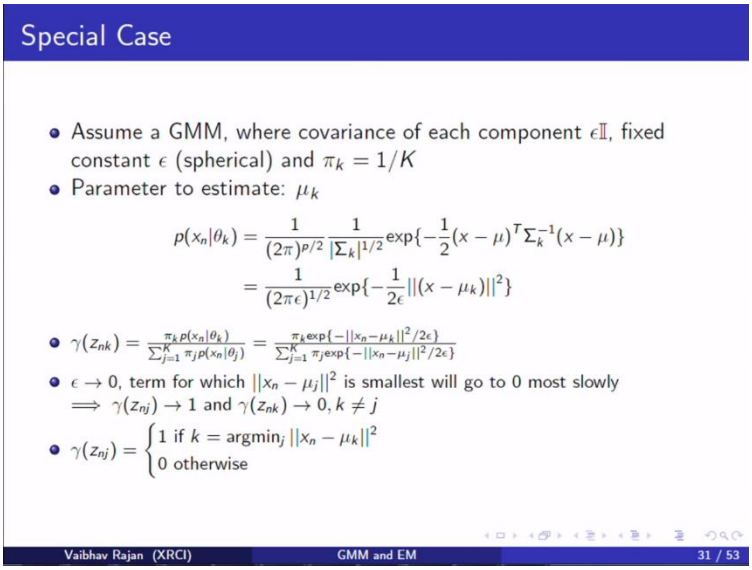
Today we are going to see that this procedure is nice because it guarantees that the likelihood will increase in every iteration. So whatever likelihood you start with, at every iteration the likelihood is going to increase. So that is what we are going to show today.

(Refer Slide Time: 06:50)



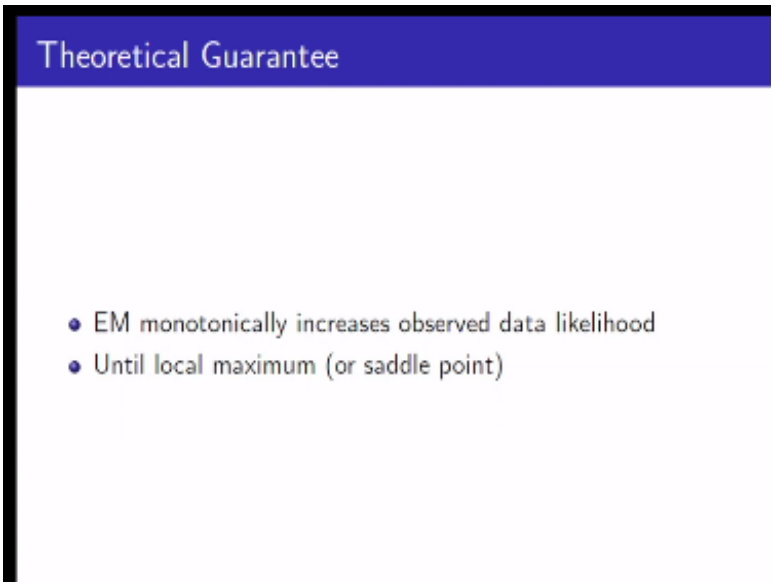
This is the complete EM algorithm for estimating the parameter of a Gaussian mixture.

(Refer Slide Time: 06:51)



And we also saw that if we assume that the only parameter to be determine is μ_k, which means we assume that all the Gaussians are spherical with known covariance matrices, and $\pi_k = \frac{1}{K}$, then essentially what we get back is the K-means algorithm.

(Refer Slide Time: 07:22)

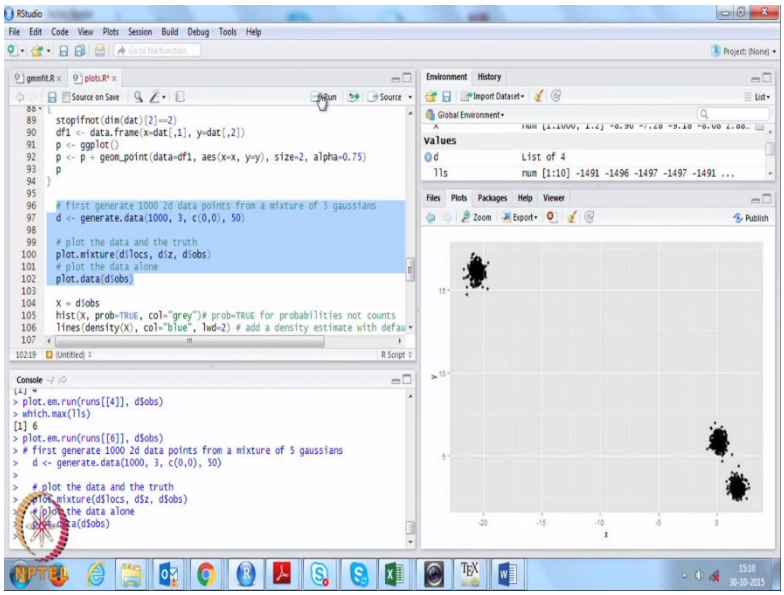


So this is the theoretical guarantee I was talking about - EM monotonically increases the observed data likelihood until it reaches some local maximum. It can also get stuck in some saddle points.

So it doesn't give you the global maximum. It only takes you to the local maximum.

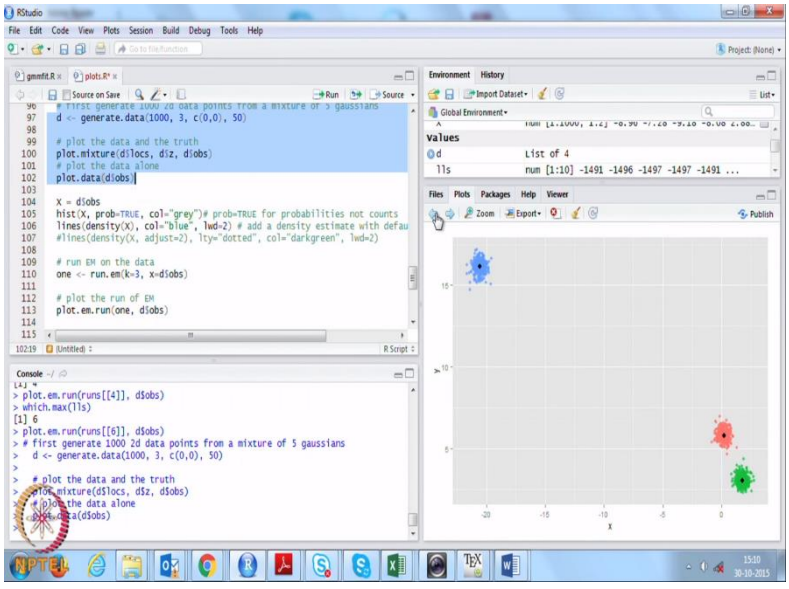
So let me show you that simulation that I had shown you last time.

(Refer Slide Time: 08:03)



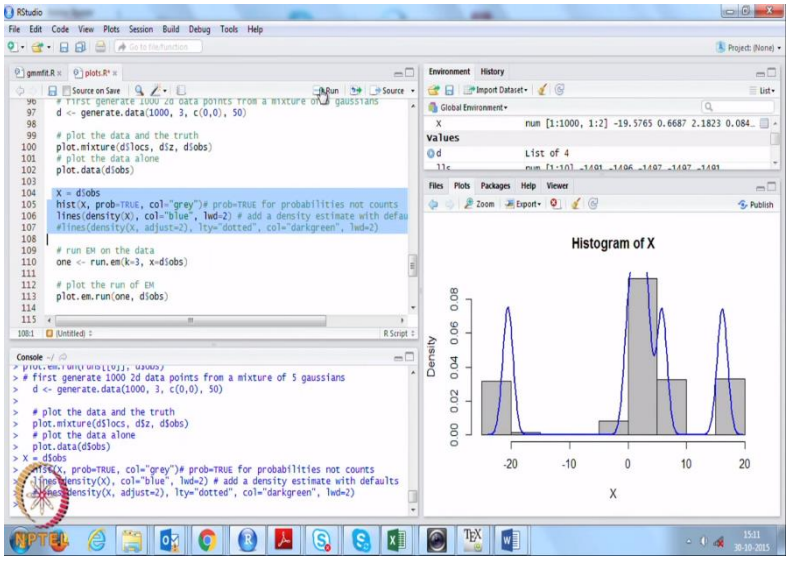
So I generate some data - 3 Gaussians. This is what the data looks like.

(Refer Slide Time: 08:09)



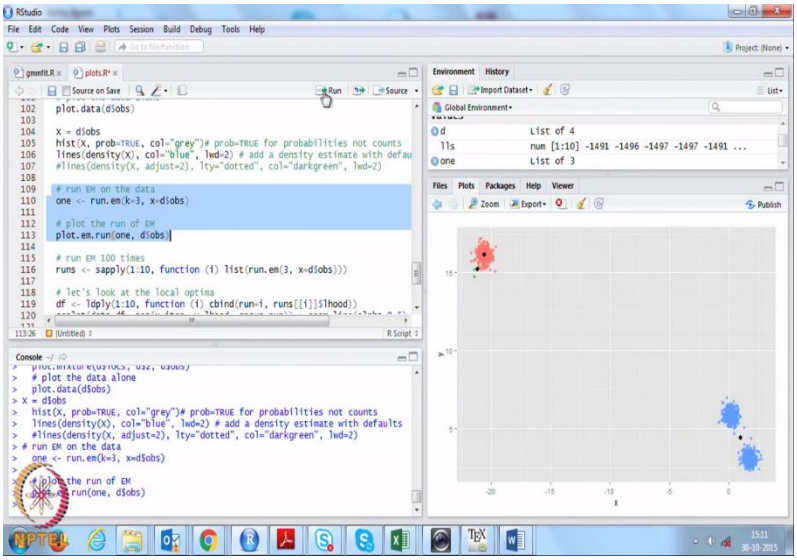
It was generated like this by taking those 3 means and covariance matrices.

(Refer Slide Time: 08:20)



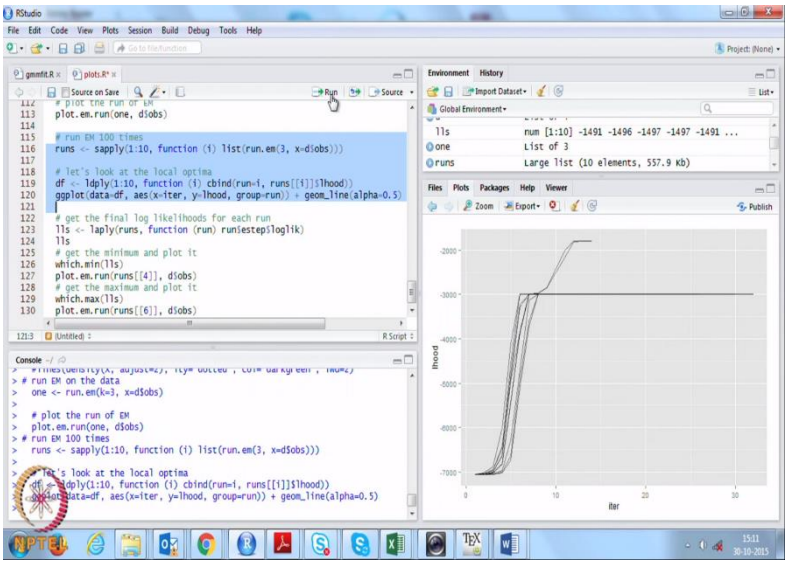
This is what the fitted density looks like.

(Refer Slide Time: 08:27)



I run EM once. So this time EM did not do well. You can see what happened. Two of the means that it had inferred are here and the third one is here. Because these two clusters are very close together it assumed that it has been generated from the same Gaussian.

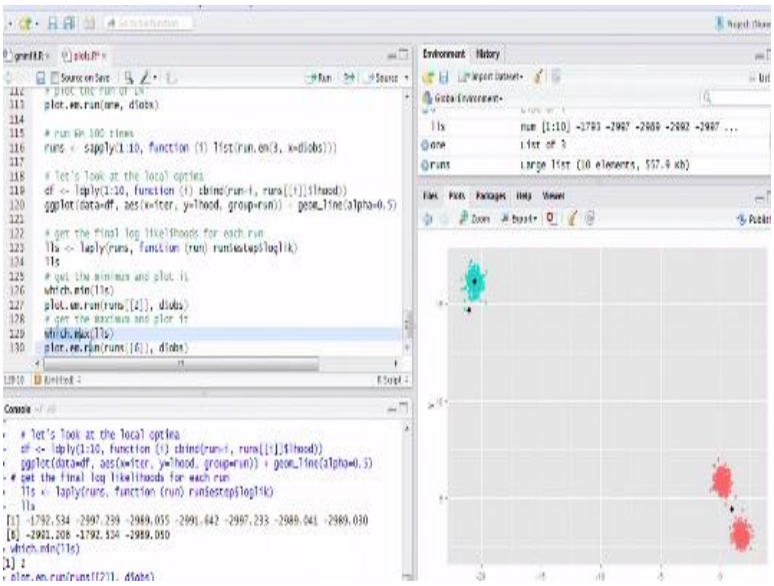
(Refer Slide Time: 08:58)



Let’s now run this. So I ran this 10 times and what I see is that for each of this run the likelihood keeps increasing.

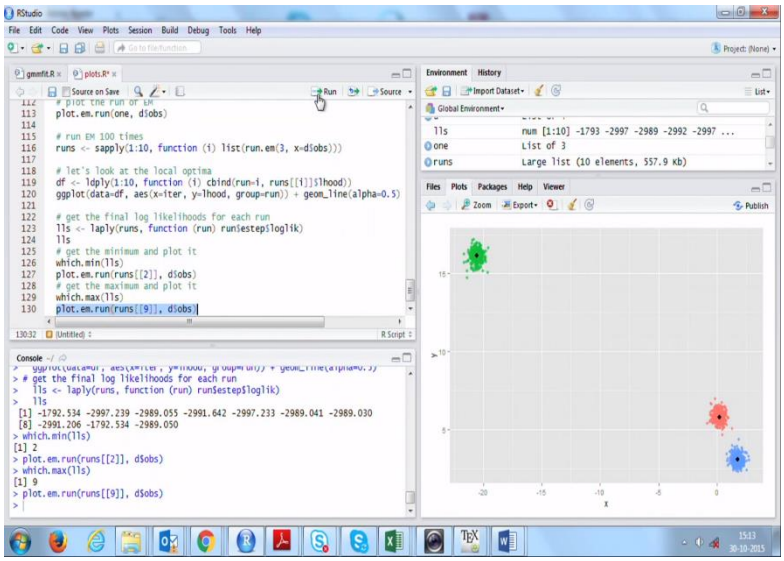
Every time, for each iteration the likelihood increases. Sometimes it gets stuck at a saddle point or fixed point and then it does not increase. So this is the typical behavior of EM. Infact this is a very good debugging tool if you are writing EM algorithms for your models. If you see that the likelihood is not increasing there is some bug in your program.

(Refer Slide Time: 09:59)



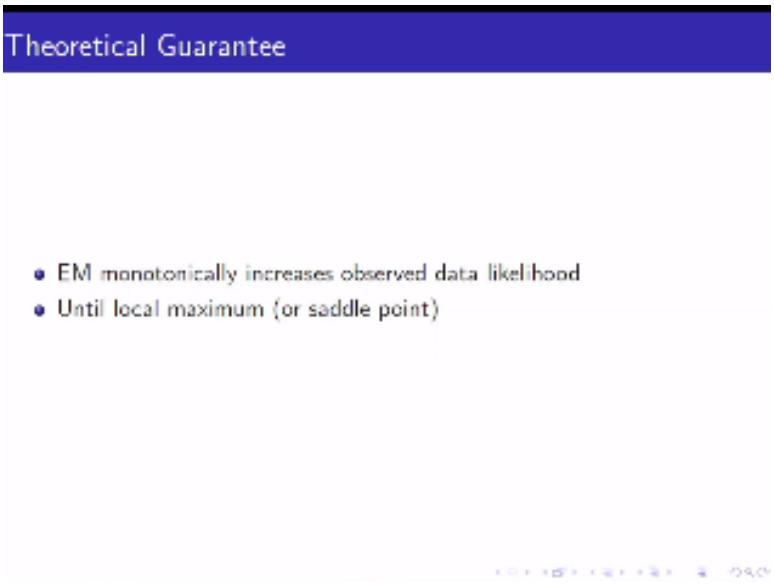
In these 10 runs, these are the likelihood values it stopped at, at the end. So if we see the minimum of these, this is the second one, and we see the fitted density when we use that run. You see the fit is not very good.

(Refer Slide Time: 10:16)



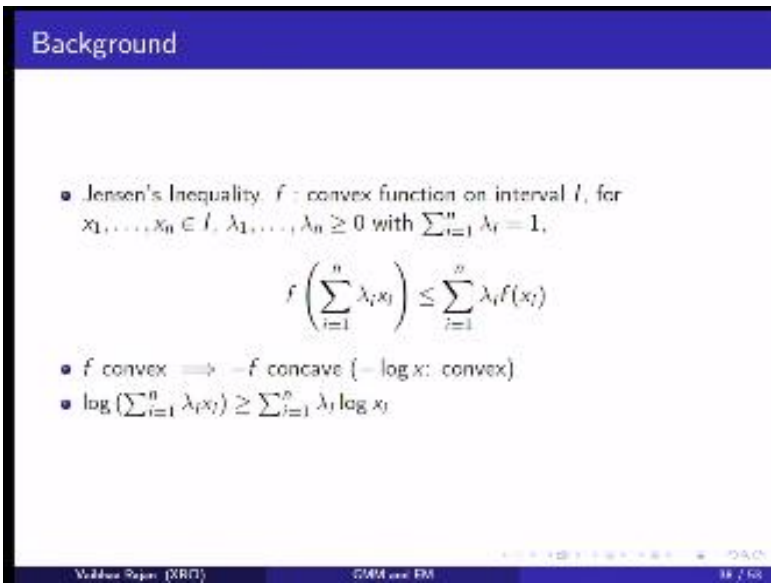
If we take the maximum likelihood among those 10 runs, it was for the 9th run. In the 9th run, the likelihood was the highest among these 10 runs and the fit was also much better.

(Refer Slide Time: 10:36)



So now let's prove that what we saw there is true in all cases, that it actually monotonically increases the likelihood in every iteration.

(Refer Slide Time: 10:49)



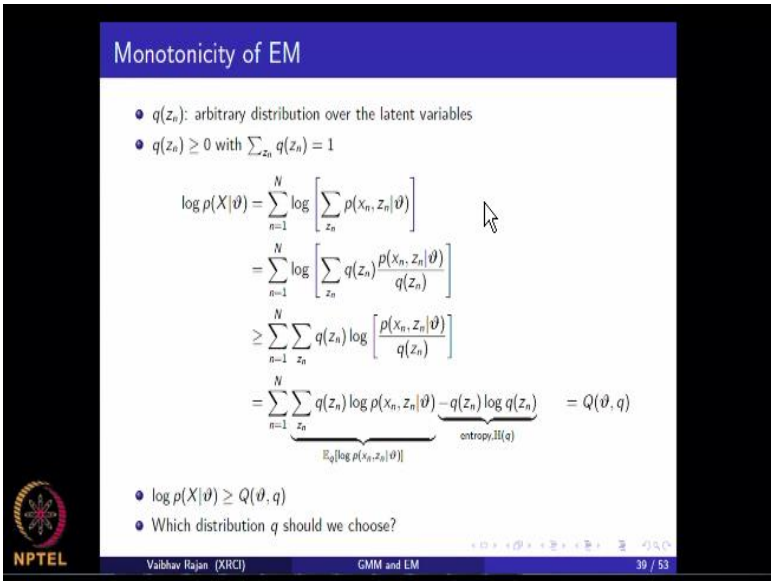
So the main result we will need to prove this is Jensen's inequality. So if you have a convex function f, and you have a linear combination of these points x_i, then the convex function applied to the linear combination is less than or equal to the linear combination of $f(x_i)$, where the function f is applied to each of the x_is.

$$f\left(\sum_{i=1}^{n} \lambda_i x_i\right) \leq \sum_{i=1}^{n} \lambda_i f(x_i)$$

And what we are interested in, as you might have guessed, is the logarithm function because log is what appears in our summation.

And if we use the fact that $-\log(x)$ is convex and put $-\log(x)$ for function f, then we get the inequality $\log(\sum_{i=1}^{n} \lambda_i x_i) \geq \sum_{i=1}^{n} \lambda_i \log x_i$. It's the same thing. The function is just the logarithm. So what we see is that log of summation is always greater than or equal to the summation of those $\lambda_i x_i$.

(Refer Slide Time: 11:59)



So we have these latent variables, or the hidden variables in some cases. Let's assume that q is some arbitratary distribution over the latent variables. We will not define what q is right now. So because these are probability values, each of these $q(z_n)$s or each latent variable is greater than 0, and the sum of $q(z_n)$ over all z_ns is equal to 1.

So now let take the likelihood of our data, and we express it again as usual in terms of joint likelihood with respect to their latent variables. And then we just multiply and divide by $q(z_n)$. Now because of this condition, $q(z_n)$ is same as λ_is here. It follows the assumptions of Jensen's inequality. So we can apply Jensen's inequality here and get a lower bound on this expression. Basically take the summation outside and get the log inside, and this lower bound just follows from Jensen's inequality. All we have done is applied this inequality.

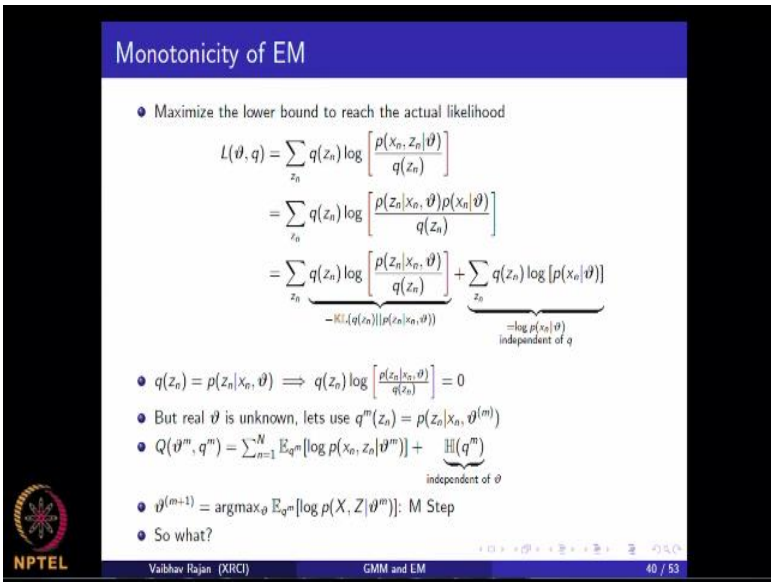
And λ_is are the $q(z_n)$s here because they are probabilities, the assumptions at true. So now this logarithm can be written as a difference of the log of the numerator minus log of the denominator. Now this expression should start looking familiar to you. This is just an expectation. It is the expectation of the complete data likelihood under the distribution q, right.

So this is something that we have been working with in EM. And on this side, we have the entropy. So this entropy term is not going to play a big role here but we are going to be interested in this. So let us call this Q. Although it will be the same Q eventually, I have used different Q here because right now we do not know that it's the same Q.

So what have we got? We have got a lower bound on the log likelihood. We have proved this for any arbitrary distribution. We have not said that it is the distribution of the latent variables under the guesses of the parameters that we had. So now the question is which distribution q should we choose. Any guesses? So what we have is a lower bound. What kind of distribution would you like to choose? No guesses? Think iteratively.

Alright, since this is the lower bound, we want the bound to be as tight as possible. So we will choose a q such that maximize the lower bound to reach the actual likelihood. So that's a natural choice when you are dealing with bounds.

(Refer Slide Time: 15:36)



So let's see how we can choose such a q. To do that let's just look at the final expression for lower bound of $\log p(X|\vartheta)$ from previous slide again. We will ignore the $\sum_n$ because we will bring it back later, but I have just not written it. So this is the original expression for the lower bound. The Q function is here. I have just written that again here. Now I am just expressing this joint likelihood or I am factorizing it in this way. So you have $p(x_n, z_n|\vartheta)$, the joint probability is just $p(z_n|x_n, \vartheta)p(x_n|\vartheta)$.

So this is just factorization of this probability. Then I just separate it out in a different way this time, and what we get here is a term which is just the Kullback-Leibler distance between $q(z_n)$ and the distribution $p(z_n|x_n, \vartheta)$. It is negative of the KL divergence between $q(z_n)$ and probability of z_n given x_n and ϑ.

And the second term is essentially summing over all z_n for $q(z_n) \log[p(x_n|\vartheta)]$. So this is independent of q, and we just get the likelihood back here, and in the first term we have the negative KL divergence between these two distributions. So if we want the lower bound to reach the actual likelihood which we are getting as the second term, we want the first term to become zero.

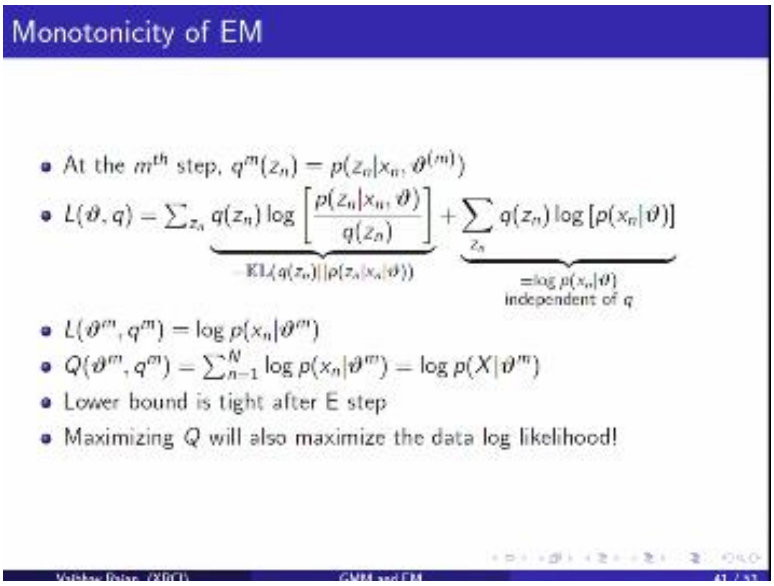
And that we can do by just putting $q(z_n)$ equal to $p(z_n|x_n, \vartheta)$. But again we come back to the same problem that we don't know the ϑ, but in an iteration of EM we have guessed the value of ϑ, $\vartheta^{(m)}$. So we can use that value of $\vartheta^{(m)}$ and use that probability distribution as q.

So what we get if we use this value for q^m, which is the probability of z_n given x_n and the guessd $\vartheta^{(m)}$ values is nothing but the expectation $\mathbb{E}_{q^m}[\log p(x_n, z_n|\vartheta^m)]$ which we saw coming up in the last slide, except that instead of q, we are using q^m, which is based on the current guess value. We are also getting entropy term but this entropy term is independent of the ϑs. So when we maximize this in our M step of EM, this does not play any role, and what we have eventually maximizing is this expectation of the likelihood under the distribution of Z.

So let me again summarizes what I did. I took the log likelihood. This is the likelihood that we are interested in for getting maximum likelihood estimates. Using Jensen's inequality, I got a lower bound. The lower bound was in the form of an expectation, which is the expectation we maximize in the E-step if we take q^m to be exactly the probability distribution $p(z_n|x_n, \vartheta^{(m)})$, and this probability distribution turns out to be exactly the probability distribution to take which will maximize the lower bound to reach the actual data log likelihood at that step.

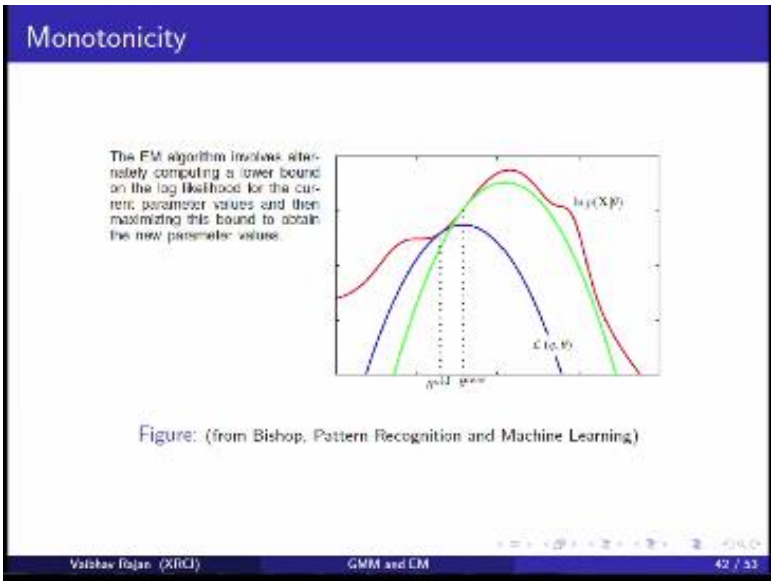
So what? What have we done? We have taken our current guesses and chosen a value of q^m that will reach the actual likelihood with respect to the current guess. But that has not bought us closer to the real ϑ. We are still working with our guesses of the parameters.

(Refer Slide Time: 20:15)



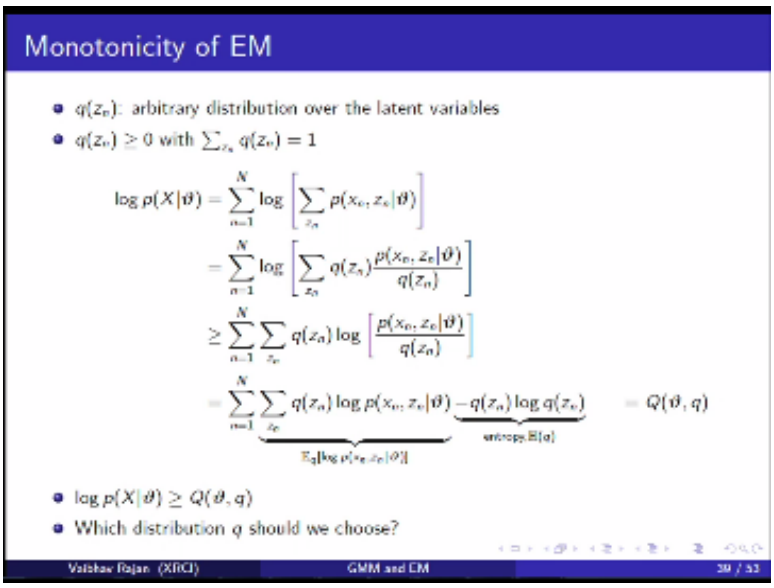
So here comes the crucial part. So at the m^{th} step, we took q^m, the distribution of z_n, to be exactly this probability distribution - the posterior distribution z_n, given the data points x_n and the current guesses of the parameters $\vartheta^{(m)}$. We saw that this likelihood is exactly equal to the KL divergence plus the log likelihood. Because this KL divergence becomes 0 at this point, this Q function is exactly equal to the log likelihood, which means the lower bound is tight after E step, which is what we wanted. So maximizing Q after this is going to maximize the data log likelihood also.

(Refer Slide Time: 21:04)



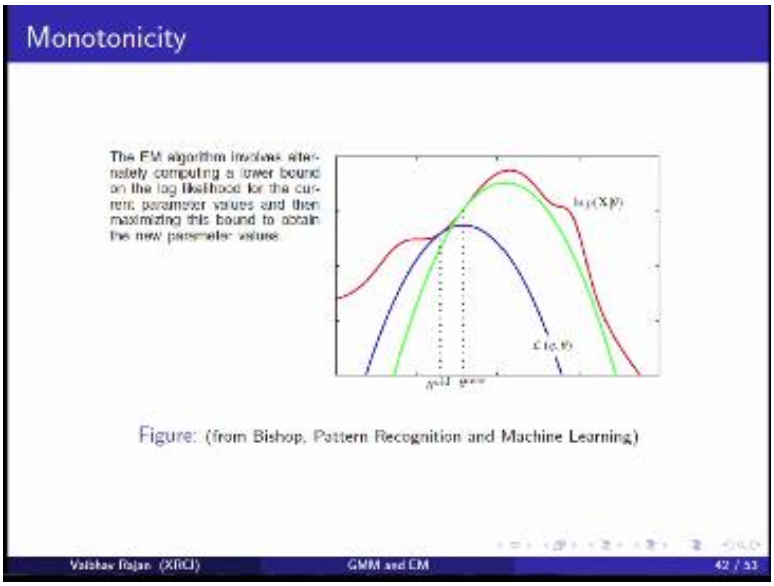
To see that, see this picture. So this is your current value, the guessed value of θ. This red curve here is the actual data log likelihood with the original parameters that you don't know. Now what the E step has ensured is that you get a lower bound using the q function that we had. So that lower bound is L.

(Refer Slide Time: 21:36)



So this is the is the lower bound, right, and which is exactly the expectation that we are trying to maximize.

(Refer Slide Time: 21:42)



So you use this L function, and you know that this is a lower bound which means it is always lesser than the red curve. The important point is that at the E step, this bound is tight which means this is touching the red curve. If you maximize $L(q, \theta)$, you will get a new set of parameters which will increase the L value but because this is touching it, and because this is the lower bound it will also increase the likelihood value with respect to the original θ.

So it is a trick because we cannot compute this likelihood, but we have computed the lower bound and we have maximizing this, but the new values are guaranteed to increase the original likihood also because at this point the approximation is tight, and we are maximizing it. So now again the at the next step, the E step will ensure that the lower bound that you calculate, the green curve, will be tight.

And once again you maximize it, you will get a value somewhere here (any way here), and the next value of θ is again going to increase the likelihood because at each step the E step will ensure

that you get a proper lower bound and you always make sure that it is tight because of the choice of the distribution of q that we take at each step.

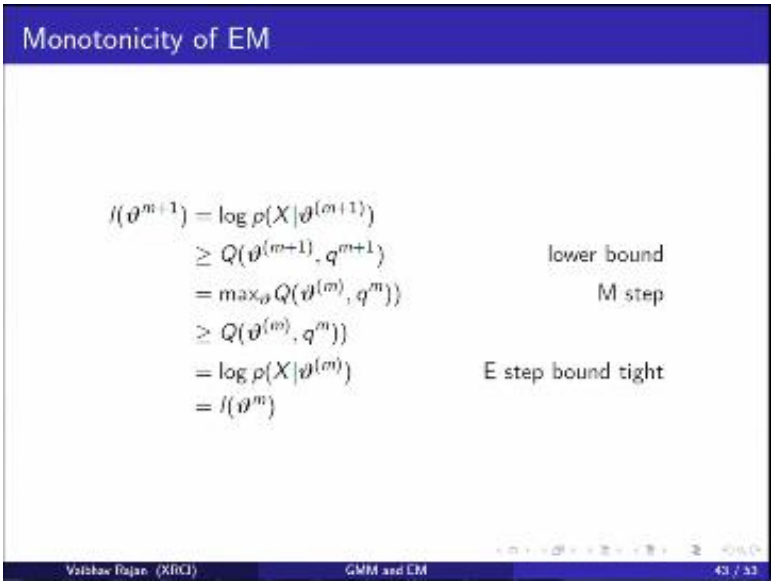
Student: Why would we get stuck in a local optima then? It looks optimal only here. Why would the expectation we reach only upto a local maxima?

Yeah because what if we get to the saddle point. The likelihood curve need not always be like this. So for example, the likelihood value can be something like this.

Student: Then why wouldn't that reach there then?

Suppose it goes like this, then at this point it is not guaranteed to go up that way. It will just be here in this region. So the usual problem with optimization. Now we can do this formally.

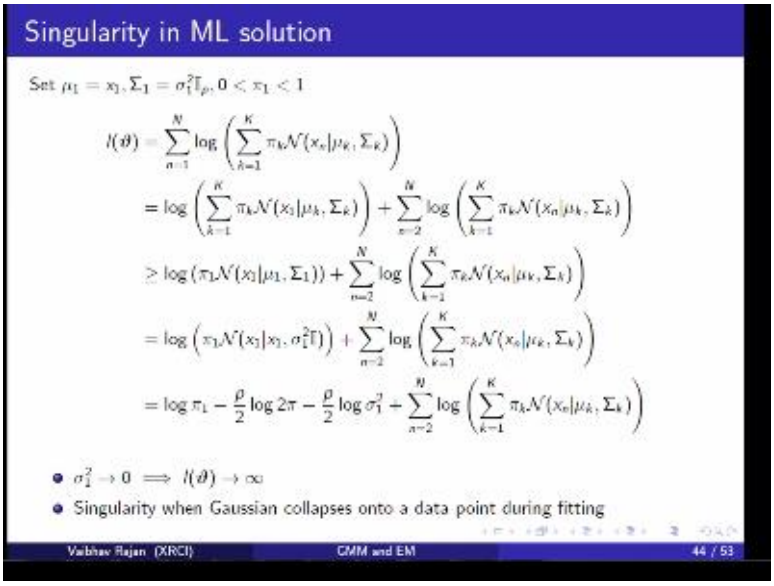
(Refer Slide Time: 24: 15)



We at the M plus first round, we have some parameters. That's the log likelihood of those parameters. And we know that Q function is lower bound. We proved it for any choice of the distribution, q. This Q value was chosen by the pervious iterations M step. So this equality follows.

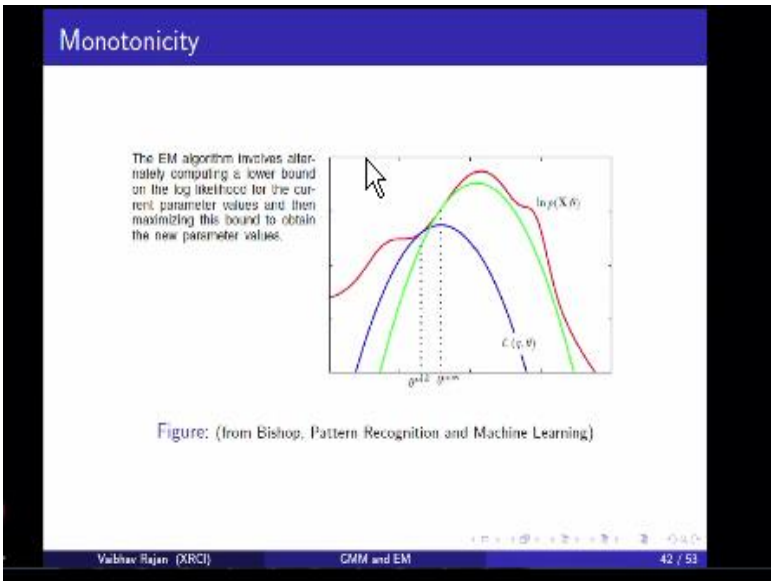
This is the maximum value of Q which is maximum over all parameters ϑ. Then this by definition is greater than any Q here, and because this E step bond is tight, we get that this is equal to the likelihood in the previous step, which is just the likelihood of the previous step.

(Refer Slide Time: 25:14)



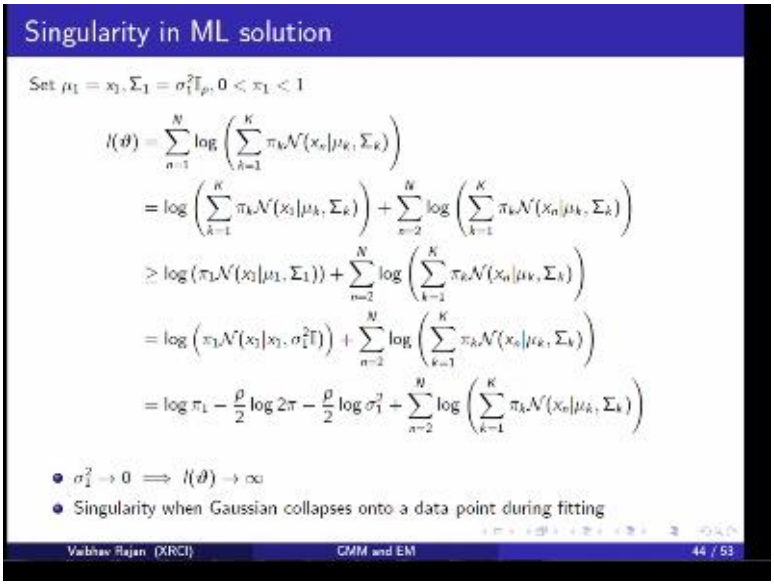
Okay.

(Refer Slide Time: 25:15)



So any questions? Now is clear why it is increasing the likelihood at each iteration. So that covers the basics of EM. Now let us look at some strange cases.

(Refer Slide Time: 25:52)

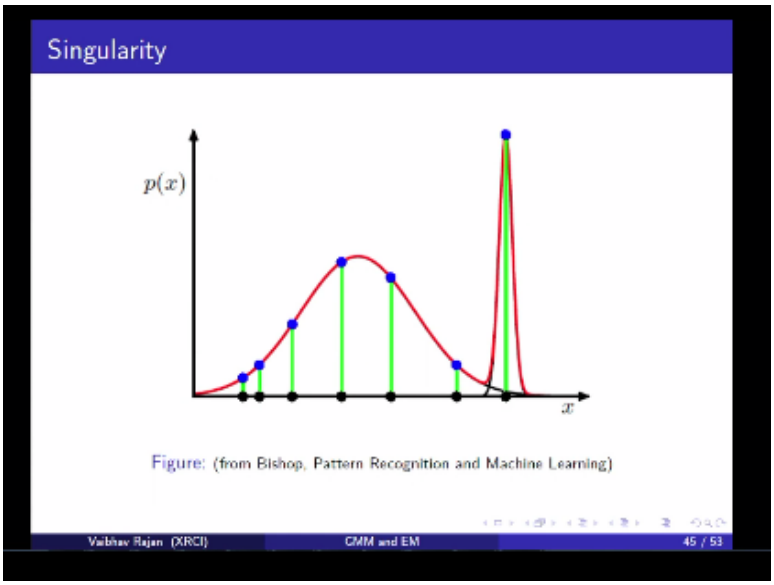


Sometimes what happens is when you are running EM, you tend to get very strange solutions and this could be one of the reasons. So I am going to motivate this mathematically. So suppose you take your log likelihood that you want to maximize.

Now suppose you have two components. It doesn't matter. So take one of the components and set μ_1 (the mean) to be equal to x_1 (one of the data points), and set Σ_1 to be equal to some diagonal matrix of dimensionality p, and take some π_1. So the expression for $l(\vartheta)$ can be just split into two parts. We are looking at just one Gaussian in the first part of the expression, and when you plug in these values of μ_1, Σ_1 and π_1, you essentially get this expression. Now what happens if σ_1^2 (the variance) tends to 0? This total likelihood essentially tends to ∞ because the value $-\frac{p}{2}\log\sigma_1^2$ goes to infinity.

So this is a problem in general with maximum likelihood solutions. The likelihood will tend to ∞ although the fit is really bad.

(Refer Slide Time: 27:10)

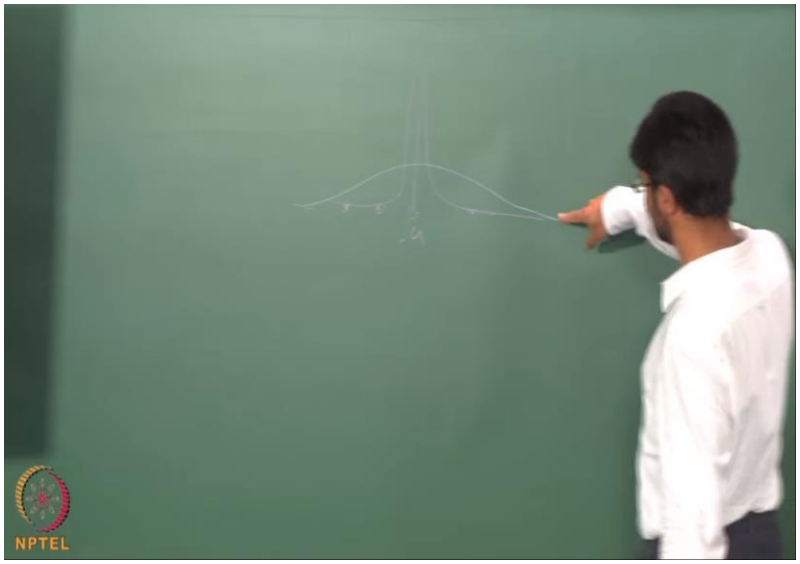


So the pictorial representation is something like this. What you are doing is you are taking two Gaussians and you are fitting just one data point with one Gaussian, and the other Gaussian is fitting the rest of the data points. In most real life cases this is not a good thing to do because it's very unlikely that the data has been generated by two Gaussians like this, with one data point from one Gaussian and the rest from the other Gaussian.

So when you try to do this with just a single Gaussian, do you think you will get this problem? Why?

Suppose I take uni-dimensional case, and I fit this one Gaussian here. This, it is, there will be a nonzero probability of a point coming from somewhere here, right. You know this is the mean.

(Refer Slide Time: 28:41)

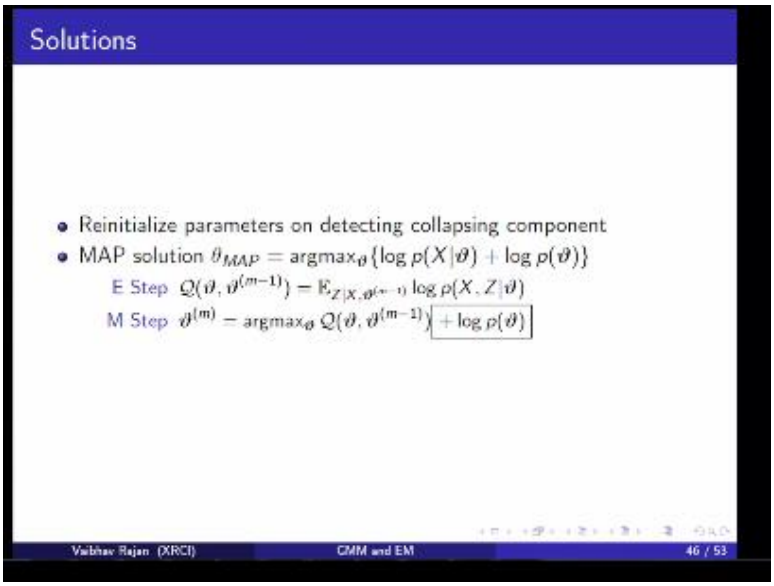


So intuitively we would think that the blue Gaussian is what might have generated this data with so much variance. But there is a nonzero probability that the data has been generated from such a Gaussian like the one in pink. So why will we not have this problem there?

So the maximum likelihood solution will never give you the Gaussian in pink. Maximum likelihood solution is most likely to give you something the Gaussian in blue. When you work out the likelihood, the likelihood for the pink Gaussian is definitely going to be lesser than the likelihood for the blue Gaussian.

Again this is just due to the mathematical form of the Gaussian mixture. Because of the summation this is really happening. Because it is possible that you can fit the data like that in a way that the likelihood goes to ∞.

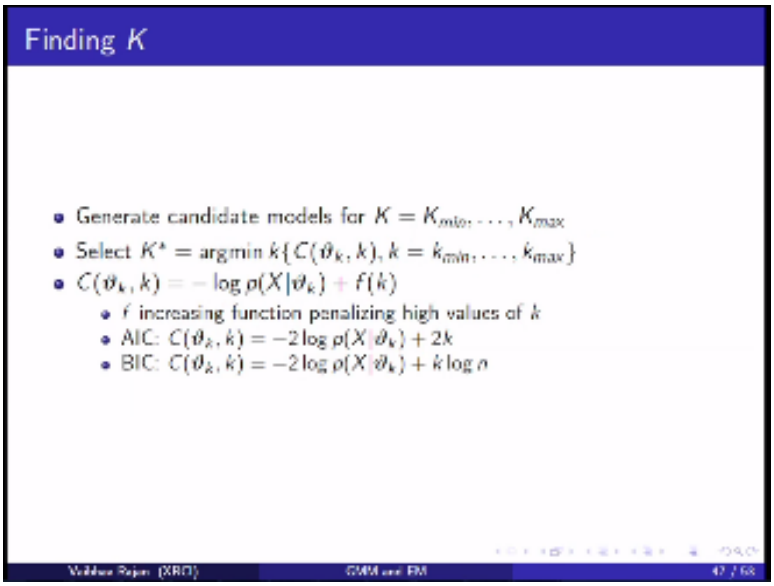
(Refer Slide Time: 29:40)



So how do you deal with this? The simplest way in a frequentist framework is when you are running EM, you check whether it is happening or not and if it is happening then you just reinitialize the parameters. You keep trying to detect such collapsing components and try to do it. And in general actually it is better to restart EM several times, because EM, as you know can get stuck at a fixed point or a saddle point. So with different initialization parameters you can get much better solutions as we saw in this simulation as well.

The bayesian solution is to take priors, right. You take priors on each of the parameters and it turns out you can work out the math and see, that the E step remains the same and the only difference necessary is the additional term in the M step that we need to maximize, and this usually solves the problem by choosing right priors.

(Refer Slide Time: 30:51)



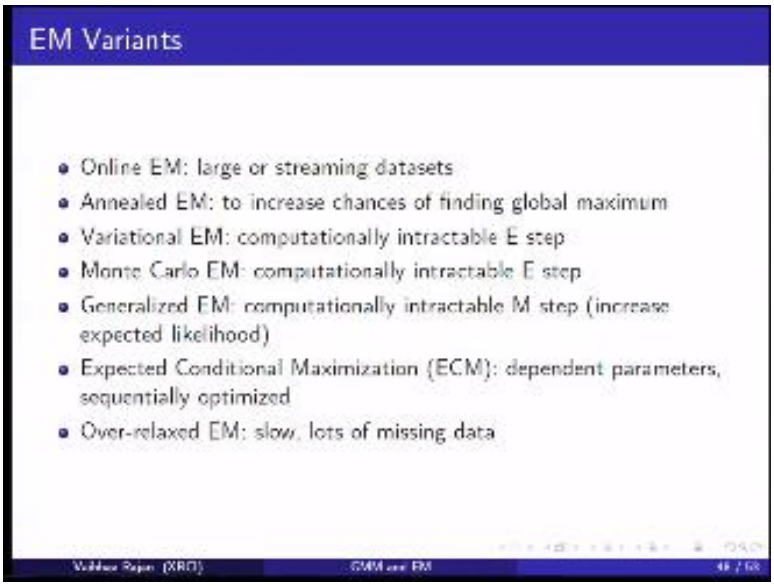
So now it's come to finding K. Till now we have assumed that we know the number of components. How do we find K?

There is no really good solution to finding K. What statisticians usually prefer and what works well in practice is to generate many candidate models. You look at the data and you assume that there can't be lesser than 3 components here and there can't be more than 12 components here. So let us run EM for all these different values of K and you choose that value of K which minimizes some criterion. There are different criteria that people have discussed. For example, it is something like the regularization that you do in another models. You basically penalize high values of K.

So the AIC, Akaike information criterion is just the log likelihood plus k. So minimizing this will give you the least number of components which can explain the data well. There is a bayesian information criterion (BIC) which uses $k \log n$ which is a similar general idea. Then there are other approaches for finding K which are bayesian nonparametric approaches where you assume some Dirichlet process priors and then the method itself automatically estimates K.

So the algorithm that we discussed in that form was given in 1977. So you can imagine that a lot of work has been done on EM since 1977.

(Refer Slide Time: 32:43)



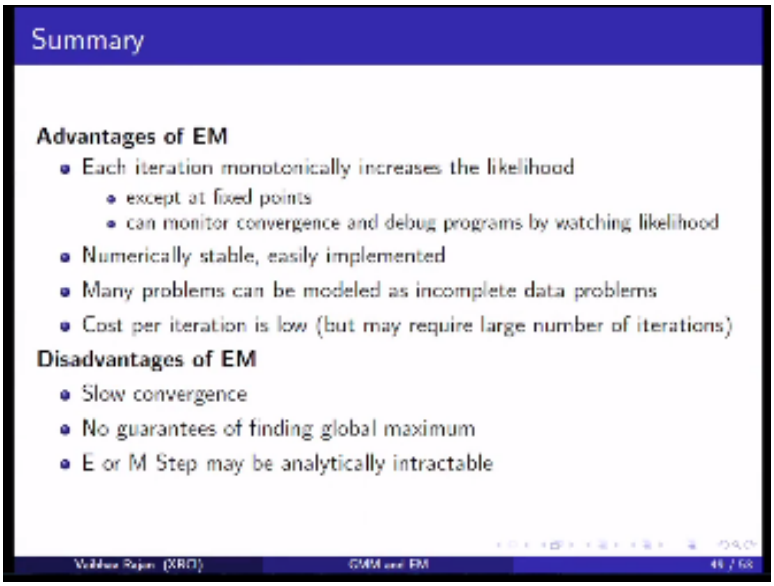
There are lot of different kinds of EM algorithms. There are online versions that work on large streaming data sets. Like I said EM is designed to find local maximum. So there are annealed versions that increases the chances of finding global maximum. The simplest solution is random restarts but annealing does something more. So in the case of Gaussian, we saw that the E step and the M steps were computationally tractable and we could derive analytical formulas for these.

But in a lot of cases, we will see that they are not computationally tractable and sometimes you need to do additional things. So there are variational versions of EM, there are stochastic versions of EM, Monte Carlo version where you have intractable E steps. There is something call generalized EM which was one of the earliest algorithms where you have computationally intractable M steps.

Then when we have sequential parameters, dependent parameters, then there are other versions of EM. In general EM is quite slow. So each step within EM, within the iteration is computationally not very expensive, but convergence is usually very slow and it's especially slow when you have

lots of missing data or lots of latent variables to infer. So there are many approaches to deal with it - these Aitken acceleration techniques over relaxed EM and so on. So to summarize…

(Refer Slide Time: 34:25)



The major advantage of EM is that it guarantees to monotonically increases the likelihood. If you take any distribution, any mixture model or any likelihood computation where there are hidden variables or latent variables, if you apply EM and if you follow the formulas carefully, you will guarantee that the likelihood is increased except at fixed points.

And it is usually numerically very stable compared to other techniques like gradient descent. It is easily implemented, and the interesting thing is that many problems can be modeled as incomplete data problems. We saw that in the case of Gaussian mixture, there is no missing data in the beginning but we assume the latent variables to be missing. The disadvantages as I mentioned is slow convergence, there is no guarantees of finding global maximum and the steps may be analytically intractable.

(Refer Slide Time: 35:24)

So the two standard references have very nice explanation for EM, and there are very nice tutorials also available. You should be familiar with "The Matrix Cookbook" to get all your matrix derivatives. and this is the standard reference if you want to go really deep into EM - McLachlan and Krishnan's book on EM. The whole book is on EM algorithm.

The EM can always solve it but it not maybe able to solve well. When there is lots of missing data then usually, it does not give good results.

So you may sometimes want to know how good those estimates that you are getting are. So you want to get the standard errors on those estimates. So in fact that is one of the flaws of EM. It does not automatically give you that but there are methods to deal with that. For example, there are some bootstrap methods that can give you error estimates for the estimated parameters.

The parameters K_{max} and K_{min} in finding K are also guessed. So that is something you have to guess based on the data that you have. So if you take the standard R packages like "mclust" or something like that, they usually have some default parameters (something like 2 and 12) but you can set them. Like what I showed in this stimulation, when "mclust" runs and tries to find the parameters, it runs it for all those different values of K and takes the best one with respect to the likelihood.

So it will be a good exercise I think if you take some different distribution, something like Bernoulli or some very simple distribution and work out the math. It will be quite nice to see how it works out.

Even the other thing that I did not work out here, the part on slide 46 is also quite simple to do. Assume that there is a prior and see how it works out. But the general idea is clear, right?

IIT Madras Production

Funded by
Department of Higher Education
Ministry of Human Resource Development
Government of India

www.nptel.ac.in

NPTEL

NPTEL ONLINE CERTIFICATION COURSE

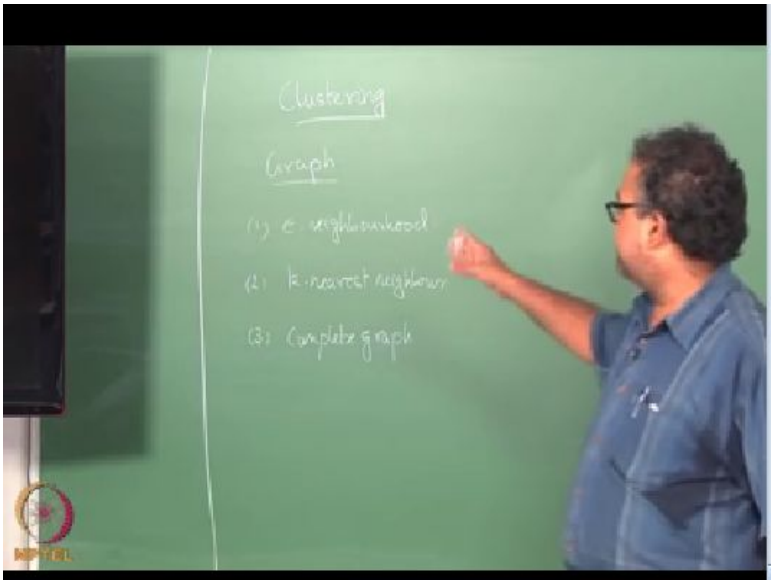
Introduction to Machine Learning

Lecture-79
Spectral Clustering

Pro. Balaraman Ravindran
Computer Science and Engineering
Indian Institute of Technology Madras

The idea behind clustering is to group data points that are similar and try to keep them as far away from things that are dissimilar etc.

(Refer Slide Time: 00:24)



1. ε - neighbourhood
2. K - nearest neighbour
3. Complete graph

A lot of convenience is to look at the clustering problem as it is done on a graph. We talked about hierarchical clustering. We could do minimum spanning trees okay and then we could do all kinds of the single link complete link clusters we could form using graphs looking at threshold graphs etc. So we looked at a variety of different things we could do with the graphs .

So it turns out that increasingly graph based clustering is becoming more and more popular we talked about density based clustering and we said it could give we kind of arbitrary kind of clusters and also we talked about cure and we said cure could also give we non convex clusters and things like that it turns out that if we use what are called spectral methods for clustering on graphs .

we get all of these in a very natural way we get all kinds of weird clusters in a very natural way because essentially we're looking at graphs here we are not really looking at any kind of metric space and as long as we represent our data in a graph we can do the clustering on it. So how do we get our data into a graph? Did we talk about constructing graph sort of data so somebody can tell me how we get this.

So the first one is we can do something like we can do what is called an epsilon neighborhood graph so what we do is we take a data point draw a circle of radius epsilon around it and any point that is there in that radius we connect it, . so this will be a symmetric relation so if A is within epsilon of B then B will be within epsilon of A so it would be a symmetric relation so we typically graphs that or constructed this way are undirected graphs.

Graphs that are constructed this way are undirected graphs and then second thing is typically graphs that are constructed this way are also unweighted graphs because epsilon is usually small and the difference the differences in the distance also will be even smaller than epsilon so we do not really want to focus on those differences in the distances we just go ahead and assign them all the same weight okay.

So it is clear what do we do with the epsilon neighborhood they still end up with and directed unweighted graphs typically. The second thing is called the k-nearest neighbor graph so what do I do? I take a point and find the k-nearest neighbors to that point and connect them . Now a question is this relationship symmetric not necessarily so A could be one of the k-nearest neighbors of B , but B need not be the k-nearest neighbors of a .

It could be slightly that could be more near neighbors that are clustered around A and so B might not fall in the k-nearest neighbor list. so typically these crafts should be directed graphs. There is an edge from A to B that means B is within the k-nearest neighbors list of a and then if there is an edge from B to a also that means that the relationship is reciprocal but otherwise not .

And also the nearest neighbors k-nearest neighbors could be very far away rather the distance need not be uniform so typically use a weight. so the most general form of this k-nearest neighbor graph should be a weighted directed graph in the epsilon neighborhood case it will be an unweighted undirected graph. In the k-nearest neighbor case it will be a weighted directed graph but for reasons of convenience and because there are larger classes of methods that operate with undirected graphs .

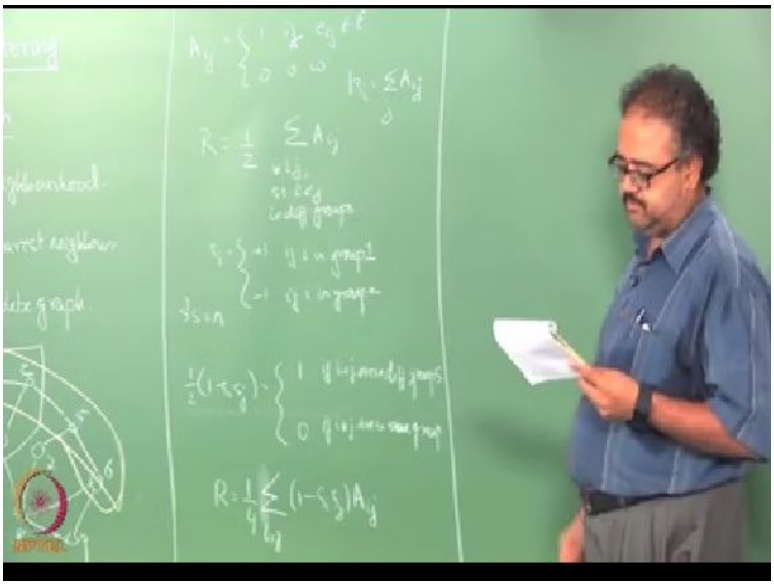
We tend to treat the k-nearest neighbor graph also as an undirected graph essentially we ignore the direction on the arrows so if there is an edge between A and B it means that either A is a k nearest neighbor of B or B is a k-nearest neighbor of a well or both its an inclusive one . so both are fine so that is essentially what, what we will do normally so we actually even treat the k-nearest neighbor graph as an undirected graph.

And so the third mechanism is we already given the graph I told we that we are just given the similarity measures between their own so we are not really given the data points and from that we can construct the graph so sometimes we essentially end up with the end up with a complete

graph so in this case we have to give weights otherwise it does not make sense so the complete graph will be a weighted graph and it just says said okay

I will connect points A and B and the weight will be proportional to or inversely proportional to the distance between A and B . so it will be inversely proportional to the distance between A and B so that means every point a and we will get connected if they are very very far away they will be connected with a very small weight but still they will all be connected okay. So this is essentially these are the three ways in which people typically take the data that we have unconverted into, into graphs okay.

(Refer Slide Time: 07:08)



So now what we do with these graphs so I will start off with a simple problem then I will try to split this into two two parts. If I have data I want to split the data into two parts okay so little some notation is an adjacency matrix.

$$A_{ij} = \begin{cases} 1 & \text{if} e_{ij} \epsilon E \\ 0 & \text{otherwise} \end{cases}$$

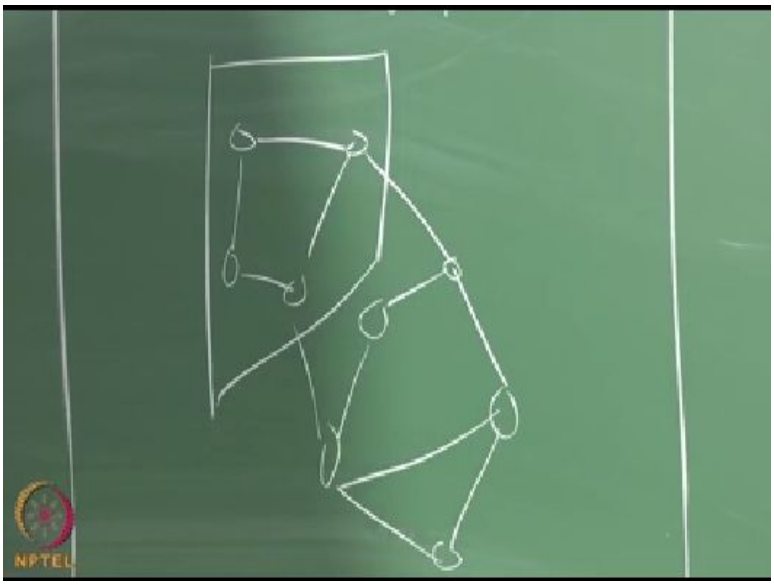
so I am going to assume that this is a symmetric matrix or time be okay and so in fact all the three could be considered as symmetric matrices . if I am going to convert all of them to an undirected graph all of them will be symmetric.

So I am just going to think of this as a symmetric matrix . I am going to define the quantity called the cut value denoted by R

$$R = \frac{1}{2} \sum_{\forall i,j}^{i\&j \text{ in diff clusters}} A_{ij}$$

So the cut value is essentially I will sum up all this A_{ij} such that i is in one group and j is in the other group so I have a graph now graph something like this let us say and I divide that into two groups that is one group that is another group .

(Refer Slide Time: 08:59)



So I am going to sum up all of these such that i is in one group & j is in the other group . let us give them numbers so that we can do some even.

some arbitrary numbers so what will I do so this sum will run over A_{15} A_{16} A_{17} A_{19} and A_{18} well all of them were meant to be 0 so we do not care so likewise A_{21} no A_{25} A_{26} A_{27} and A_{29} so only A_{25} will be 1 so I will count that once and then for 3 everything is 0 I do not care and 4 I will get the 1 I will count the once .

And what then? Do I stop ? I keep going so 5 I will count it so what will happen 5 again will get a 1 so I will count it once so 6 I do not care so 6 will be what so A_{62} A_{61} A_{64} A_{63} so all of them 0. Likewise I keep going and for 8 I get a 1 okay so how many do I count? I count 4 . accounted four but truly how many edges have I cut? 2 and that is why the half okay

so this way I do it. When I do this I count every edge twice and some more notation .

$$s_i = \begin{cases} +1 & \text{if i in group 1} \\ -1 & \text{if i in group 2} \end{cases}$$

.

So I am saying S_i is +1 if I am in group 1 is -1 if i is in group 2 just an indicator variable okay. so one thing just remember that for all less S^T S will be n where n is the number of nodes okay so n is a number of data points so S_i is the i^{th} entry in that vector S okay which indicates whether the i^{th} data point belongs to group1 or group2

let us look at this expression

$$\frac{1}{2}(1 - s_i s_j) = \begin{cases} 1 & \text{if i\&j are in diff groups} \\ 0 & \text{if i\&j are in same groups} \end{cases}$$

So when will be 1?. sorry both in different groups . they are both in different groups it will be 1 both in the same group that will be 0. so if i and j are in j are in the same group with me 0 if they

are the different groups it will be1 here okay. Just a bag and not a person loud enough to be a person sorry now this just the assignment so we just take some assignment when we are trying to evaluate it.

So like I said I have arbitrarily chosen this assignment that I cross in a different assignment like I could have said 1, 2, 5, 6 or in one cluster one group and the remainder or in another group and P could have evaluated that okay. Let us why do not we do that for fun so what will be the R value get 5 now R value if I so I can just take any clusters any grouping like this and I can evaluate and find R value .

So my best is to find the R value that is the smallest and would be the smallest all of the degenerate solutions to this problem when all the nodes belong to group one or group to so the cut will be zero. So we have to avoid the degenerate solution. We will see that in fall out naturally from here a degenerate solution will come out as the best solution so we have to explicitly exclude great.

So far so good, so what are we doing here? We have essentially come up with a mathematical expression that captures this such that i and j are in different groups so instead of saying that I could just multiply the terms here by this and some over all ij. If they are in the same group the number will get added if they are in different groups I will get a zero so instead of doing something number some like this.

That can do something more compact weight so I can write R as.

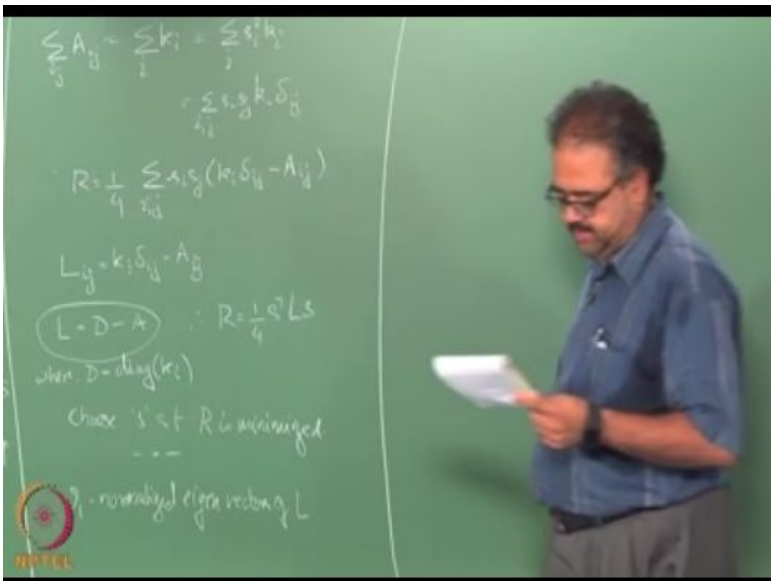
$$R = \frac{1}{4}\sum_{i,j}(1 - s_i s_j)A_{ij}$$

okay so that half I have another half here so they get a ij a run over 1 to n okay. so one other notation I should introduce

$$R_i = \sum_j A_{ij}$$

what is that? degree if we want to get the degree of node i just sum over j A_{ij} so that will give me the degree of node i. So what will be the first term here? just Aij is some over that will be the first term for R okay.

(Refer Slide Time: 16:51)



I am just going to take that and rewrite that into something more complex. What is the sum over ij A_{ij} ? twice the number of edges but we will write it a little more complex form so that we can go back, plug it in and derive one nice expression okay.

So this I will write it as a summation over i of k_i.

$$\sum_{ij} A_{ij} = \sum_{i} K_i = \sum_{i} s_i^2 k_i$$

$$= \sum_{ij} s_i s_j k_i \delta_{ij}$$

can i do that? dah !

The here (Δ_{ij}) is a function that is 1 if i = j 0 otherwise sounds like this but there is a purpose for doing it but all of us buy this. but if we are wondering why (S_i^2) square is okay ($S_i^{2\text{) square}}$ is just 1 okay so I am just multiplying it by 1 so it is fine and then I am just splitting that into $S_i S_j$ and writing it into a more complex form.

$$R = \frac{1}{4} \sum_{i,j} s_i s_j (k_i \delta_{ij} - A_{ij})$$

Now I am going to substitute it back there so essentially what I have done here is I have produced $S_i S_j$ on both terms .

so I am going to introduce a new matrix L_{ij} which so the ijth at entry of the matrix L will be this so if we think about it . I can write this as D-A where D is a diagonal matrix where the i^{th} entry in the diagonal is the degree of node i .

$$L_{ij} = k_i \delta_{ij} - A_{ij}$$

So this expression is sometimes called the laplacian is called the laplacian of the matrix A is also called the un-normalized laplacian of the matrix A so the normalized laplacian we actually do a transformation on this and has got better properties in terms of both in terms of clustering and also in terms of other things which people use the laplacian for but I am just going to show we how to work with the un-normalized laplacian.

In form the actual algorithm for working with the normalized laplacian is exactly the same as working with un-normalized laplacian except that proving things are slightly different showing that we are getting a good clustering is slightly different so I will give we material to read up on both normalization and un-normalized laplacian but we will look un-normalized in the class .

$$L = D - A; where, D = diagonal(k_i)$$

So this makes sense so the laplacian is D-A and it is not an arbitrary choice we arrived at it by trying to say that I am looking at the cut size and I want something that characterizes the cut size then we did a lot of algebra after that we ended up with D-A it is not the only way to derive the laplacian there are many ways in which many different fields in which people are actually independently come up with something that looks like the laplacian.

And then it is got wide applicability

so how many actually use laplacians before of graphs

this not it not in matrices okay I am not yeah it is very closely related to our laplacian in the continuous domain actually

$$R = \frac{1}{4} S^T L S$$

so we do over matrix notation transformation so I can say that might cut this just 1/4 th $S^{Transpose}$ LS so S_i S_j x L_{ij} . So essentially $S^{Transpose}$ LS

So now the goal is to choose S so that R is minimized.

so the the development I am following comes from Neumann who is one of the pioneers in looking at graph partitioning and what the social network people called community direction and

soon so forth, righ.t I mean he did not come up with a spectral clustering spectral clustering is older but so Neumann came up with this very nice way of introducing spectral clustering .

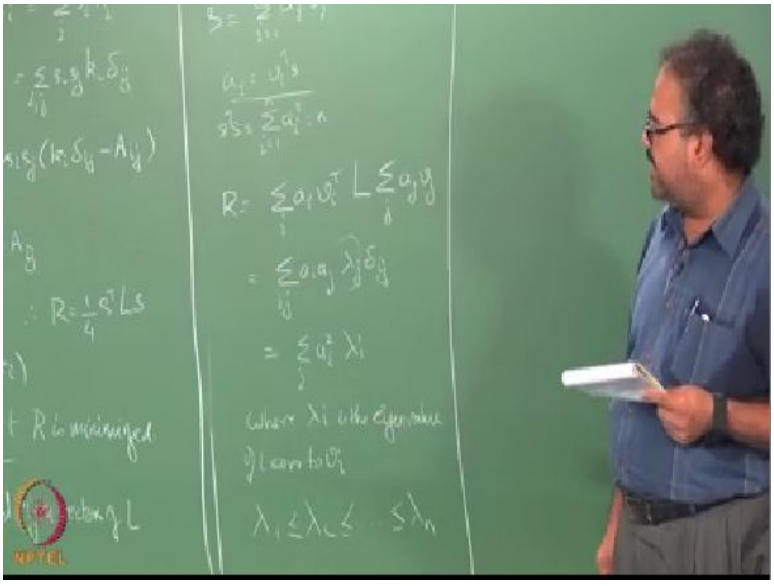
Without ever going into real analysis or functional analysis or anything just going with little bit of algebra and a little bit of linear algebra so far we are not even done linear algebra except for that very rudimentary transformation so just looking at very basic concepts he kind of motivates how we do the partitioning that if we typically go read other things they will start off with say okay.

Here are the properties of the laplacian and we apply these properties and therefore we can solve this like in one short and here is an answer and things like that so it becomes little more tricky so I am going to make we read all of that but I just want we to appreciate that the idea behind this is fairly straightforward so whatever we are doing is sadly straightforward so now

so now we will enter into the spectral domain . I want to say Vi denotes a kind of a normalized at

Vi denotes the normalized eigenvectors of L .

(Refer Slide Time: 25:55)



I take the S and I can it as

$$s = \sum_{i=1}^{n} a_i v_i$$

$$a_i = v_i^T s$$

$$S^T S = \sum_{i=1}^{n} a_i^2 = n$$

so I have I have eigenvectors that they are going to span my n-dimensional space so I can just take any point S which is in the n dimensional space I can write it as combination of the eigenvectors so we about it A_i is just essentially and s'f'

not sure why I need this anyway yeah

so note that $S^{Transpose}S$ is what n and $S^{Transpose}S$ is what sum of $A^{2\ squared}$ so that should be equal to n that is a constraint that we should have in mind.

So now let us go back and make the R look complex again

the let us $A^{Transpose}$ S so here comes R the fact that V_i and these are eigenvectors of L so what would be L will Vj λ jvj so I can just write it as λ jvj and what about $V_i^{Transpose} V_j$ Δ_{ij} $A^T V_j$ will be 0 i is not equal to j so essentially what I will be left out with is.

then I mean I Could have written λ i λ_j which over it does not matter.

We had to write λ_j because this stick with this notation that I had the j on the -hand side i is fine?

so this is essentially equal to

What is Aij?

So S is a vector so I am trying to express S in the coordinate space defined by the eigenvectors of L so essentially what I do is take a look at the projection of S on each of those dimensions and then write it as a sum of those so that is essentially what Aij is.

This is again basic linear algebra the only only place where we use the spectra is when we went from here to here
so I am using the fact that my VS are going to give me a basis like and in fact we are giving me an orthogonal well they are giving me an orthonormal basis because I am assuming they are normalized so giving me orthonormal basis and that is why I get this Δ_{ij} here and then the only place where I used it is the introduction of this λ here λ_j here.

$$R = \sum_i a_i v_i^T L \sum_j a_j v_j$$
$$= \sum_{ij} a_i a_j \lambda_j \delta_{ij}$$
$$= \sum_i a_i^2 \lambda_i$$
$$\text{where} \lambda_i \text{is the eigen value if i corr. to } v_i$$
$$\lambda_1 \leq \lambda_2 \leq \ldots . \lambda_n$$

So this is K so why I can collapse the summation back into this great

Now what do we do so what is λ here?

so I am going to assume that the s λ are indexed such that λ 1 is the smallest Eigen value λ 2 is the next highest and so on so forth okay.

So now if I want to minimize this expression all I really want to do is minimize my R now found to minimize the expression so what should I do?

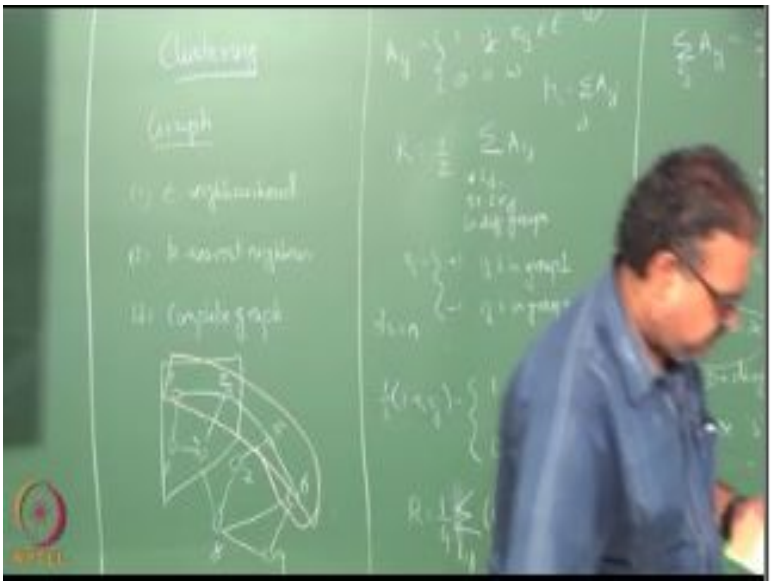
So I should choose my A such that the maximum weight is scale placed on the smallest λ. In fact I should place all the weight of the smallest λ and what is all the weight?

so we have that somewhere so summation A_i $^{2 \text{ squared}}$ is n so I need to keep all the weight on what is the smallest value that is λ 1

What is λ 1?

what is that if we are wondering what that symbol meant it was L.

(Refer Slide Time: 32:52)



So then they summed up one row of the laplacian so what do I get ?

$$\sum_j L_{ij} = \sum_j (k_i \delta_{ij} - A_{ij}) = k_i - k_j = 0$$

Guys come on.

so this will be what k_i totally listening? okay!

so each row of the laplacian would sum to 0 and if we think about it what is the diagonal element the degree of that node and all the off-diagonal elements or negatives of the edges so whenever there is an edge there will be a -1 there is no edge there will be nothing so the diagonal entries degree and then I will have as many -1 as they are our neighbors.

So adding up the row will give me 0

likewise adding a column will give me

symmetry when we are thinking so long symmetric okay so it should give we the same thing great

so now what does this really mean the fact that the row sum is 0 what does it mean all one is an eigenvector with Eigenvalue of 0

and with a little bit of additional work we can show that this is the smallest .

Because its laplacian this is symmetric and it also turns out to be positive semi-definite and that for all the Eigenvalues will have to be non-negative so we can we can easily verify that L will be positive semi-definite

so how to verify it will be positive semi-definite

take an arbitrary F so $F^{Transpose}$ L F should be greater than or equal to 0 so if we can show that for any arbitrary choice of F that will happen then we are all set.

So it is for L is positive semi definite that for 0 will be the lowest Eigenvalue so all we need to do is make sure that we put all our weight on the first Eigenvalue okay we are all set

so what is our first eigenvector in this case and so by V_1choice is actually normalized so it will be divided by 1 by root n (1√n) 1 by root n (1√n) 1 by root n (1√n) 1 by root n (1√n) will be the eigenvector that I am choosing for this in this case so how will I minimize this I essentially have to align my S with this eigenvector so I will get the maximum A_1 correct.

We will see the thing so what is hey one what is the A1? A1 is V_i^T S so if I want all the way to go to a1 essentially all I need to do is choose sin the direction of V1 and what will be S? what will be the inner product of s with any of the other v is 0 so if I choose it to be in the direction of V1 I get my assignment but choosing it to the direction of V1 means what? I assign it to all 1s this is exactly the degenerate solution we are talking about .

So taking s to be all 1s means I am putting everything in group 1 at all I can put everything in group group2 does not matter one of those things so choosing yes to be all ones essentially is a degenerate solution I was selling it avoid so what we should do is we should say that I am going to exclude this because this is a degenerate solution so find an S for me that is orthogonal to all 1s .

Because I have to exclude this direction I have to look at the space spanned by the rest of the eigenvectors so that space will be orthogonal to the original is this dimension that we are excluding so essentially I have to put an additional constraint I am not only interested in minimizing R but I want a minimizer that is orthogonal to the all ones so we want actually exclude this solution .

So there are many ways in which we can do this one simple fix is to say that I will fix

fix the sizes of the two groups to n1 and n2 now fix the sizes of the two groups to n1 and n2 so essentially now I am saying find two groups that minimizes the cut size say such that one group

as n1 elements other group has n 2 elements so if there is some prior knowledge that we have that lets we decide on this n1 n2 grade.

Otherwise what we can do is we can start off by saying okay. I am going to assume I wonder 50-50 split okay and then search around that to get a better thing or we do not have to do this at all we can try something else okay which I will talk about now just a second so once we have excluded 1 what is the next thing we can possibly try to do align with v2 I try to put all the weight on $\lambda 2$.

Because λ 1 is excluded forest so we have to go to λ to try to put all the weight on λ a 2 so this is also called the Fiedler vector. So v2 essentially the λ the Eigenvector corresponding to the second smallest Eigenvalue of the laplacian is called the Fiedler vector so what we try to do is we choose our s to lie in the direction of V2 okay can
restricted to + ot - 1.

So s is already restricted rate this can be have only either + or - one entries so I cannot really choose a our S arbitrary early just to lie in the direction of V2 so I have to look at this pace of + or -1s and figure out which is the closest to V2 too so essentially what I want to do is

I want to maximize

$$\text{maximize } |v_2^T s| = |\sum_i v_i^{(2)} s_i| \leq \sum_i |v_i^2|$$

I want to maximize that
S that is either +or -1 so I can get this so essentially the maxima will be achieved with this is the maxima .

we cannot get better than this the maximum will be achieved when I am making sure that these signs are all positive shape basically signs are all same so we'll make sure that signs are all positive or all negative so that is essentially what I have to ensure here so what I do now is then I look at V2 write and so this is essentially the higher component of V2 I look at the add component of V2 if it is greater than 0.

I make SI + 1which lesser than 0 I make SI -1so that is basically
so now I have a way of dividing it into two groups and hopefully okay I have a sufficient number that are plus 1 and a sufficient number that are -1 therefore I do not end up getting everything in one class or the other usually we will find that that is a very nice in fact yeah I will put up some materials on the moodle .

So where we can see pictures of the the eigenvectors it is what what does well what does v1 look like what does V1 to look like and so on so forth so what would V1 look like straight line v1 will be a straight line assuming that our graph is connected if we graph as multiple components so what will happen is our Eigenvalues 0 will have a higher multiplicity then 1 .

Rates of Eigen so the number of components in our system number of components in our graph essentially is given by the multiplicity of the Eigenvalue 0 and the Eigenvectors like the first Eigenvector will essentially be an indicator function saying which node belongs to the first component the second Eigenvector will be an indicator function that says which nodes belong to the second component .

So all the Eigenvectors corresponding to the Eigenvalue zero will essentially be indicator functions of which components in the graph their underlying thing belongs two sessions will be something like this one for all this way and 0 elsewhere and then next one will go like this for a while and then it will be 0 elsewhere the third one will be so it will be like that so so so that we can we will see that so it is not that the first eigenvector will always be flat .

If we have multiple components we actually see indicator functions there okay and in that case how will we do clustering? well our components already give us clusters the only tricky part is now within the clusters within the components. Do we want to do clustering in which case we will not actually look at the appropriate Eigen vectors so essentially we could either say that okay I am going to take the designator component separately.

$$s_i = \begin{cases} +1 & \text{if } v_i^{(2)} \geq 0 \\ -1 & \text{if } v_i^{(2)} < 0 \end{cases}$$

And then find out the Fiedler vector there can then assign it within that component

before somebody asked me what we do for equal to zero just put it somewhere else great and the second λ 2 write is also sometimes called the algebraic connectivity of the graph it tells we how well-connected the graph is and then things like that there were a whole bunch of other things associated with the interpretations associated with each of the Eigenvalues of the laplacian it.

But the second one is the most important one okay great so so far we have been talking about dividing it into two so what if we want to do more than two what is previous kind of clustering methods but this requires one possible nothing we take the adjacency matrix okay the whole thing will reduce to this take the adjacency matrix okay compute the laplacian compute the Eigenvectors of the laplacian do not even compute all the eigenvectors of laplacian.

And I want we to compete only the second eigenvector of the laplacian so there are incremental methods that actually give we the laplacian one at a time instead of actually doing the I mean the eigenvectors one at a time instead of doing it completely their incremental methods which can

work on very large graphs and then give we the eigenvectors one at a time where we just compute the second eigenvector .

And then do this that is basically it so this is step one step two well I do not have a proper step three anywhere step three is essentially finding these finding the spectra and step four is this now if we are doing it naively finding the Eigenvectors of the laplacian is a very expensive operation so that is that is a problem so one is not cheap step one is not cheap when it is a one time computation.

we do that at the beginning so we do that at the beginning is not iterative it is it at the beginning we can store this side so if we have one some of the other methods also become little easier but not k-means because in K means we have an arbitrary point somewhere even it in every iteration we have an arbitrary point which is centroid and now we have to find the distance to the arbitrary point.

So we will not be able to minimize this but here we are not introducing any arbitrary points only the fixed end points that we have and we can find the distances between the fixed end points and use any one of those methods and get my adjacency matrix and once I just do a this is this is fairly trivial this is fairly trivial and then after that I do this this is again time-consuming but once I have done that this is again fairly trivial and I get the clusters okay.

So the advantages are it tends to give we a lot more varied cluster I mean there is no restriction on the shapes of the clusters that we are finding so we can find all kinds of different shapes in the classes there is no implicit assumption about that and that turns out to be incredibly robust to small noise and things so there are lots of nice properties to spectral clustering .

And therefore this one of the reasons is becoming more popular nobody says lots of tools and packets are available it is not like we're doing multiple passes over the data in fact over the data we will do one pass when we construct our A_{ij} after that is all operations with the adjacency

matrix so does not matter what how large dimension our data was in we do one pass over the data or well however cleverly we want to organize it in we do one pass over the data or well however cleverly we want to organize it we fiddle around with the data once and we get Aij now that becomes a dimensionality of the data so data could lie in a P dimensional space .

So P could be far larger than n
but if will be working only with the N dimensional data typically
when is far larger than P

sorry image data yeah an image data if P is for far larger than in is it? yeah
depends on how we count p and in suppose yeah but they are typically Pis far larger than Nin working with image data so in many of these domains where we have this kind of structure spectral clustering is a lot more helpful .

And I also got a lot of cool theory it is we in fact exactly where we are approximating even though we cannot tell by how much we are approximating it by at least we have some kind of understanding of what is going on and K means well we have EM. to guide we there but so we need the EM version of k-means ray so we have to get we there but it still it is a rough approximation .

So all of this is confusing but we really have to do only these three steps to get our clusters so one thing I forgot to mention here before I go on to multiple clusters

so what if we have the sizes fixed to n1 and n2
So how will we do then so we can just do this essay equal to plus one if it is greater than or equal to 0 and -1 if it is less than zero we cannot do that .

I might have more in I might not be able to match then n1 , n2.
so what do I do in that case

so I start off from the most positive end and I keep marking everything as 1until I reach n1 and the remainder I say is n2 which is the cluster 2 will make sense
start off I order the points from most positive to most negative Eigenvector component start at one end I start at the positive most positive end and I say everything is +1 until reach n1 points and then the remainder will say is – 1.

Alternatively I can start from the most negative end I can keep going until I reach n1 points now not n2points that's the something okay until I reach n1 points and then assign the remaining n2 to plus 1 these are actually two different ways of splitting the graph but if I had gone from the most negative point and gone up to n2 and then assign the remaining to n1 well that will be the same but if I choose the n1 either from the top or from the bottom.

So I will actually get two different split points I will get two different clustering so which one do we pick?

which one which of whichever one has the lower R
so now that I have a method of comparing I just compare our which one has the lowest R
I will pick that ok
so this Vi2 is there so I will order the most positive Vi2 to the most negative so we know what is VI2 the Ith component of the second Eigenvector.

So I will take the component wise and I will sort it so that the most positive is at the top and the most negative is at the bottom
okay
so how do we do multiple clusters
one way to do it is to repeat divide the two clusters divided into two now I have two graphs so once I have done the separation I have two graphs and in each of those graphs I can go and divide it into two again can I reuse the same Eigenvectors whatever thing I have done no typically no so I that essentially reduce everything all over again.

I would have changed even though a lot of the connections are still the same would still be having some sub matrix preserve I will still have some sub matrix pressured but because I have chopped off some of the entries so I have to redo the computation again so I sensed have to redo this again find the fiedler value for that reduced graph not proceed okay instead a satisfactory solution may be not.

When I say divided in two I mean we might have made choices which we do not want to-do should have made a different choice if we say divided into four in the first place not divided into three in the first place so 3 is even worse so what would we do if we have three clusters so which one of the two will we split into two so I start off by dividing something into to and then I want three cluster so I have to pick one of them.

So I can pick the larger one and divided into two but that not my that might not be the choice then we can argue saying that kind of pic divide both into to whichever gives we the better cut value take that and leave the other one as a whole cluster we could do the thing of all kinds of heuristics to do this so typically this repeated division done when we are looking to split it into some nice powers of two .

And quite often this is something which people use when they are looking at VLSI layouts when they are trying to look at lawets on the chip and then they want to do some kind of segmentation of the circuits that they want to put on the ship but they typically end up doing something like this so the reason they want to do this is so I want to put two things far apart on the chip I better have few wires going cross .

So these are the line sat I am cutting so they want to keep this as small as possible the number of wires longer wires I have to draw they want to keep these things as small as possible So they essentially do this kind of repeated clustering and they use spectral clustering a lot in that so

what people do is I am just going to tell we what people are not going to actually explain why that is a

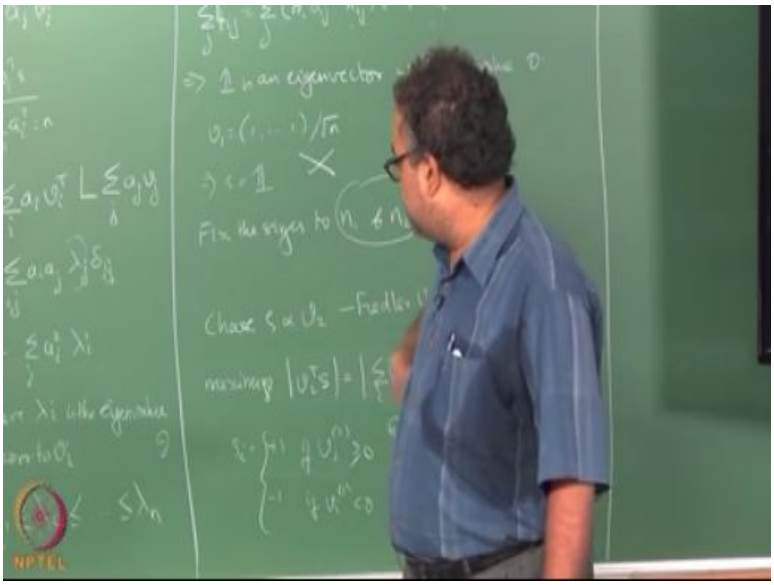
good what way of doing it I am going to let we read it up so what people essentially do is

they do k-means clustering I want Clusters okay they do k-means clustering but in a different space

what they do is they take the data points they do all of these things and then they compute the top k eigenvectors.

If they compute the top k eigenvectors and each data point each node in the graph that we had originally will get transformed to a point in a k-dimensional space so what would that mean suppose I have data point I data point I will be represented by v1i, v2i, v3i all the way to Vki a so essentially I will be arranging the eigenvectors and then reading of throws to get my data points so is it clear ?.

(Refer Slide Time: 59:04)



So I will essentially be doing this so V 1 V2 The column vector say okay this is actually a matrix so v1 is v1 runs like this we runs like this V2 runs like so then the first row here is essentially a

representation for data point 1 the second dose are presentation for data point 2 the third row is a representation for data point 3 and so on so forth so I take all of these things and then run k-means on that space.

It turns out that if the graph originally had good separation it is the data itself was originally nicely separated into K clusters if there is some way of separating them into K clusters then the data points in this projected space that will be well separated I will very clearly see k groups so it is easy for me to recover this with k-means so k-means gets confused if we have all kinds of if our representation is not is not proper so

so the usual example is this.

That is one okay i forget how the other one starts but let us do it this way that is another so our density based clustering methods like DB scan will return this as one cluster and that as an another cluster but k-means will completely mess-up k-means will do something like okay this is one cluster and that is another cluster or something very weird .

But then what Happens in the spectral domain is that when intake these data points and then try to project them into my the eigenvector space they actually project into different parts in the space these points will project to somewhere here these points will project to somewhere there and I can run by K means very easily and recover these data points.

I they occur these clusters sorry so that is essentially the idea behind doing multiple clusters with the spectra and
so this is where the un-normalized normalized business becomes important so far we didn't have to worry about the difference between normalization normalized and there are different definitions for the normalized laplacian so one definition is essentially it is whatever D power minus half meets people number d so d is a diagonal matrix so L is this one.

So we take this is one definition for a normalized laplacian is also sometimes called the centered laplacian and then the other definition I believe is sorry

$$D^{-1}L = (I - D^{-1}A)$$

I which is essentially I - D inverse when I- D inverse A. Whatever I was written here so the first one is called the centered laplacian the second one is called the random walk laplacian.

So why is it called the random walk laplacian think of what d inverses means is for every entry where if there is an edge between that node and J that entry will be 1 by degree of I so if I am doing a random walk from I that is a probability with which I will take that edge so I look at all my neighbors it will be taking edge with equal probability so that is what the random walk test so 1by DIJ 1 by Di will be the probability with which I will take any particular edge therefore the second thing is called the random walk laplacian and then they use that also .

And apart from the definitely difference in the definitions of the Laplace jeans the algorithms are typically the same so we construct the adjacency matrix find the Laplacian find the eigenvectors then if we are looking at two clusters we do this if we are looking at multiple clusters we form that matrix a so that will be an N by K matrix so form that and then we do-means on the data points and the only thing that will change is which laplacian we are using throughout

IIT Madras Production

Funded by

Department of the higher education

Ministry of the human resource department

Government of India

www.nptel.ac.in

NPTEL

NPTEL ONLINE CERTIFICATION COURSE

Introduction to Machine Learning

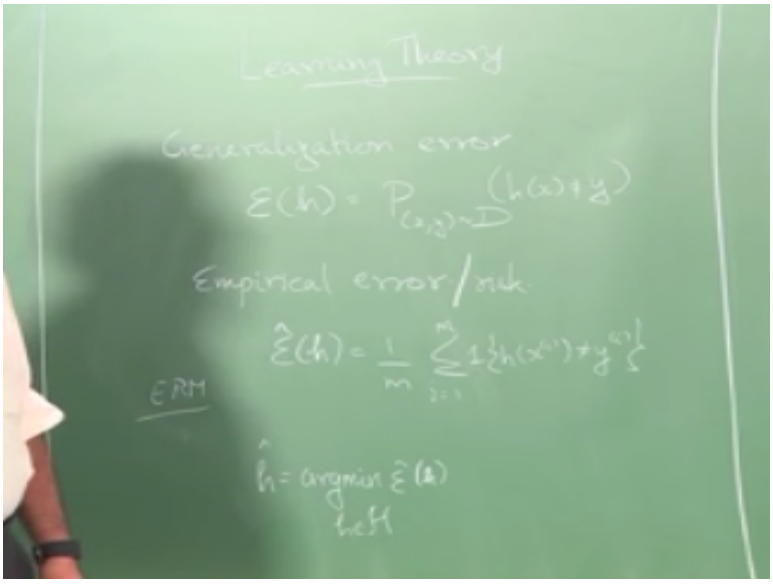
Lecture-80
Learning Theory

Prof: Balaraman Ravindran
Computer Science and Engineering
Indian Institute of Technology Madras

Computer science folks, all of you are aware that theory in computer science means talking about hardness of a problems , so how hard is to solve and looking at space complexity, time complexity , and approximability . How approximate is the solution, those kinds of things . So we are going to try and see if we can give such a flavor of theory to machine learning as well .

So we are going to talk about how hard is the problem to approximate . So I have a perfect solution , but then I can get close enough to it , but how often can I get close enough to the solution. So that is the one kind of question that people want to ask. And then there is another question on how hard problems are to solve in general. So we are going to talk about hardness of problems. So different measures of the hardness of the problems .

(Refer Slide Time: 01:16)



So typically we are interested in generalization error.

So what is generalization error?

So ε(h) is the generalization error of a hypothesis H, so I will denote my ε the error function, and of h means the error of hypothesis H .

$$\varepsilon(h) = P_{(x,y)\sim D}(h(x) \neq y)$$

The error of H this is the probability of the hypothesis H making a mistake on a data point X. When the data points X and Y, where X is the input and Y is the label, so where the data X and label Y are sampled from some underlying distribution D. If you are assuming that this distributing D is fixed, apriori and unknown we remember about all of this .

So we always talked about this, I talked about the $P_{(x, y)}$ earlier, but here we are talking about the distribution as D. So when the data is sampled according to this distribution D, so what is the probability that I will make a mistake ? So this is also the expected number of mistakes, because every mistake will count once. So this is essentially the generalization error.

But typically what is the error that we have access to ? We have access to something called an empirical error or sometimes denoted as empirical risk, which we will denote it by ε^.

$$\varepsilon\hat{}(h) = \frac{1}{m} \sum_{i=1}^{m} 1\{h(x^{(i)}) \neq y^{(i)}\}$$

So where xi and yi are the i^{th} data point given to you in the training set , and 1 is the indicator function. This function will be 1 if this condition is true it will be 0 otherwise

So essentially when it will be 1, whenever I make an error, whenever h makes an error this will be 1, whenever h is correct this will be 0. So I will add it up for all the training data. I am assuming that I have m samples in my training set , I will divide by m.
So this gives me the probability of me making a mistake as estimated from the training data . So this is sometimes known as empirical error or empirical risk.

So typically I only have this quantity. So whatever is given to me as a training data. I can have many ways in which I can estimate this error, but this is all I have access to. So this is called the empirical risk. What I mainly interested in this is the generalization error.

So what we want to know is how good is this empirical risk estimate that I make in terms of measuring the generalization error, or how close is it to the generalization error. This is the question that we want to ask. See this has shades of hypothesis testing. So we are going to do a very different kind of analysis here. It has shades of hypothesis testing, but the kind of analysis we do here will be very different.

So the question that we are asking is, given that I can estimate empirical risk, what can I say about the generalization error. So before we go on to look at this in more detailed, I want to introduce a couple of results which would make easier for us to talk about. Before that let me talk about one thing. So most learning algorithms that we have, do what is known as empirical risk minimization .

So the answer that you will typically end up giving is,

$$\hat{h} = \text{argmin}_{h \in H} \quad \hat{\varepsilon}(h)$$

so you will have some hypothesis class H, suppose you are looking at linear classifiers and your input dimension is say some P. So then you will be essentially looking at classifiers that are defined by θ_0, θ_1, θ_2 or β_0, β_1, β_2 up to β_p and then given by the inner product of that with the data point. And if it is greater than 0 you will classify it as one class, if it lesser than 0 you will classify it as another class. That is basically what linear classification does.

So when we look at the variety of linear classifiers, at the end of it you will have something like that. You will have some $\beta^T x$ and then you will have some function of $\beta^T x$ and then whether that is greater than 0 you put it with one class, if it is lesser than 0 you will put it in the other class. So that is essentially what the hypothesis class H would be. That would be in the case of linear classifiers.

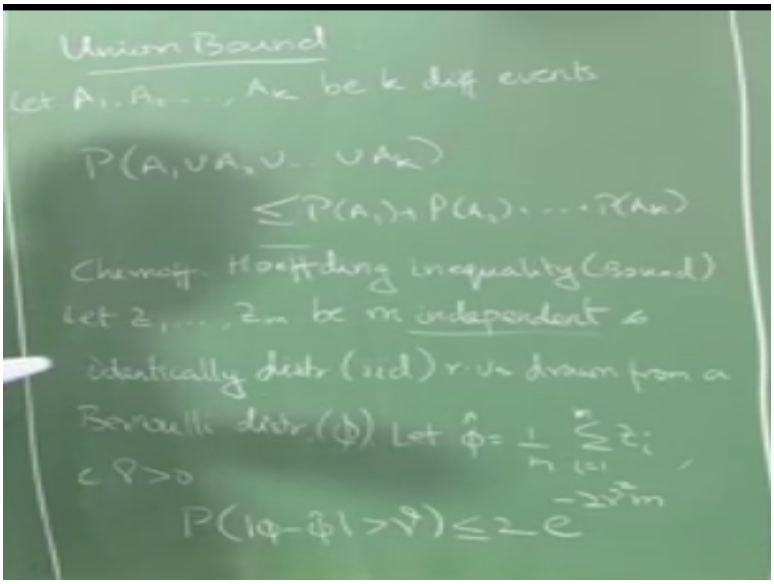
In the case of neural networks, it will be all the classifiers that we can implement given the choice of number of layers and number of neurons per layer that we had. When I make some number of neurons, some number of layers choice, so for all those different values you can set for all those weights you will get a different classifier. That constitutes your hypothesis family subscript h. So what I typically would like to report is that h, has the minimum empirical error over all members of that family H. And that $\hat{h}$ is the classifier that we will report by doing empirical risk minimization. This is essentially called empirical risk minimization.

When this is ideal case, obviously we know that we do not get this. If you are using neural networks or training using back propagation or neural net and stuff like that, we actually do not find the argmin , you essentially have some approximation of it, then you just stay with it.

So likewise, depending on what classifier you are using you might not actually find the minimum. What you are trying to do is minimize the empirical risk anyway, because that is the only thing that we can measure. So we would really like to get a classifier that is good according to this, but we are not able to do that.

So we look at empirical risk, and then we find empirical risk minimization and we get this. So before I move on, I want to introduce a couple of results.

(Refer Slide Time: 10:07)



So the first one is called Union bound. Let A1, A2, ... to Ak be k different events, then ,

$$P(A1 \cup A2 \cup \ldots \cup Ak) \leq P(A1) + P(A2) + \ldots + P(Ak)$$

In most versions of probability theory this is taken to be axiomatic okay. So this is called the union bound.
So it is equal when, they are independent or they are disjoint. If we are thinking of them as sets they are disjoint.

So this is something which is variably called a Chernoff Hoeffding bound or the hoeffding inequality or the Chernoff inequality, when some subset of these two and some subset of that will be used for describing this result.

Let Z_1, Z_m be m independent & identically distributed (iid) , drawn from a Bernoulli distribution.

So here I am stating this very specific to Bernoulli distribution but the Chernoff bound holds for in general.

There are other milder conditions, it need not necessarily be Bernoulli but as far as we are concern we are only interested in binary outcomes.
So can you guess what is outcome I am interested in ? Correctly classified or not correctly classified. So that is the outcome I am interested in so we are only considering Bernoulli in this case. In fact there is a version of it where you can also relax the independent assumption but gets more and more complicated.

In fact when you relax independent assumption you get some kind of the chromatic number or the interrelationship graph enters the picture. I still have not figured out how the chromatic number enters the picture, it gets really complicated lets so let us hope and pray that all the random variables we deal with out independent or you can think of this like that. But in some case that is not true you will have to worry about it.

So in this case essentially what I mean is, the probability that some Zi equal to one is say some ϕ .

The probability that $Z_i = 0$ is 1- ϕ. So I will just keep it as some Bernoulli distribution parameterized based some ϕ.
So what do we know about the Bernoulli distribution? ϕ is also the mean. We know the ϕ is also the mean. So typically the Chernoff Hoeffding bound is stated on the mean. But this version of the Chernoff Hoeffding bound is stated on the parameter ϕ. But it is not in the role as the probability, but the ϕ is here is used in the role of mean.Because that is one thing I want you to remember like I do not want you to look at the Chernoff Bounds and then go actually flip through some other place and find that nowhere is the probability of outcome is mentioned, that will never be the case because Chernoff Hoeffding bound is stated on the mean.

Bernoulli distribution (ϕ), Let $\hat{\phi} = \frac{1}{m} \sum_{i=1}^{m} Z_i$

So what is the $\hat{\phi}$? It is the mean estimated from these random variables I mean this all be familiar to you from the hypothesis testing case this ϕ is the true mean of the distribution and $\hat{\phi}$ is the mean that is estimated from these random variables the samples that I have drawn.

Right and $\gamma \geq$ zero, there is some fixed constant $\gamma \geq$ zero. So the probability that $\hat{\phi}$ is away from ϕ greater than γ. So what does this mean $\hat{\phi}$ is away from $\phi \geq \gamma$, is,

$$P(|\phi - \hat{\phi} | > \gamma) \leq 2e^{- \gamma 2m}$$

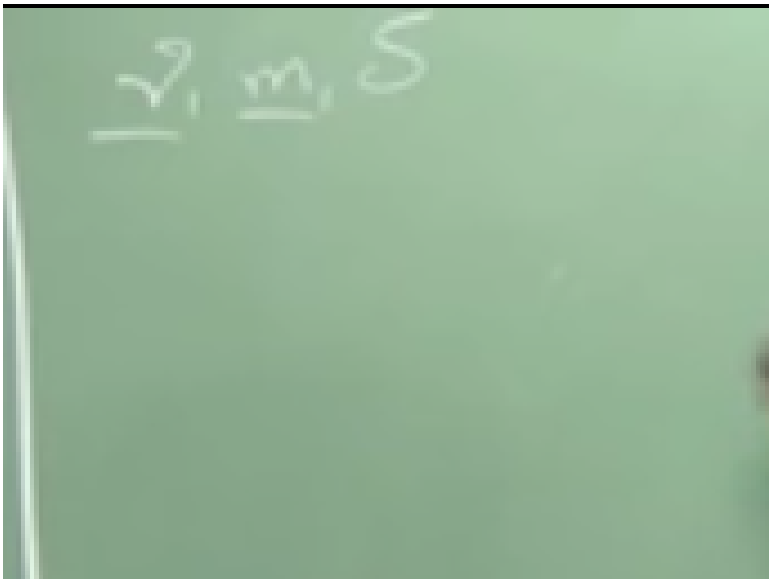
The 2 comes because this is a 2 sided inequality.
If you think about it, it is less than γ or greater than γ. It is two sided inequality therefore the 2 comes so if you want to look only at 1 side then you can drop the 2 and the use $\mathbf{e}^{- \gamma 2m}$.

So essentially what does it say if I have lot of samples? If m is very large , I have $e^{- \gamma 2m}$, so it becomes smaller and smaller, as m becomes large. The probability that my $\hat{\phi}$ will be far away from ϕ becomes smaller and smaller. So this gives you a way of quantifying how many samples I need before my estimate that I am making. Before the mean that I have estimated is close enough to the true mean, with the high probability.

So γ is something I fix apriori. γ is something, I need you to be at least this accurate for me. Now go and tell me how many samples I need.
Alternative I can ask a question like, I have so many samples, how accurate I am likely to be ? Is it fine, is it enough to fix the number of samples ?
You have to be little bit more work.

So what is the probability that the error I am making is greater than γ ? So I need supply γ. An error for me to find m. I need to supply m and an error for me to find γ.

(Refer Slide Time: 19:53)



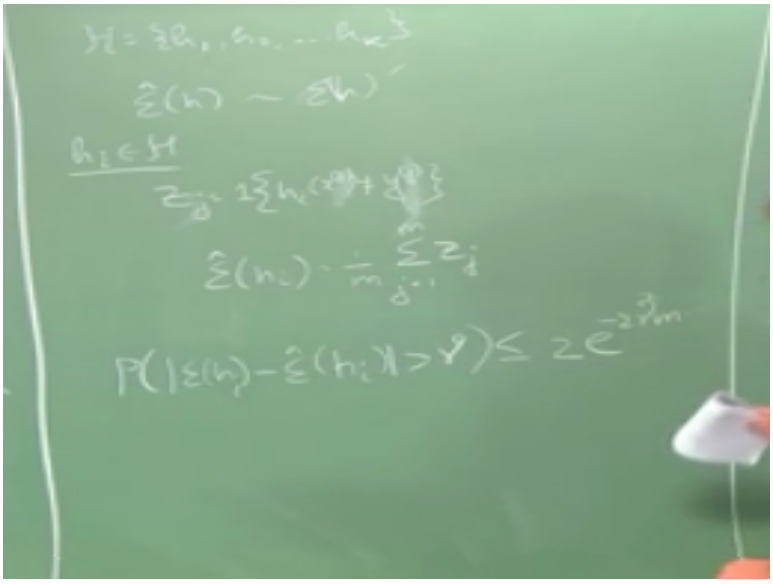
There are 3 quantities here. So there is γ, there is m and there is also the probability of the error.

So γ is the magnitude of the error. What is the probability that I am greater than γ. So that is a 3rd quantity. So I have number of samples, I have the magnitude of the error and then I have the probability of making an error of that magnitude. That is the left hand side. So I have 3 things here. So this equation has 3 things here.
So if I want to solve for γ so I need to specify the left hand side and I need to specify m, then I can solve for γ.

I say, I do not want to make a mistake more than 10% of the times. That means my probability should be 0.1 . I am only giving you 100 samples, then I will come back and tell you to be correct 90% of the times then you will have to say that even if I am so far away from the hand side I am correct. Okay only then I can give you the 90% guarantee. So that is essentially what I means by saying there are 3 things. Here you have to specify any two then you can think about it in terms of the 3 one.

So now we have these 2 results for us. So usually that probability is denoted by sum ∂. okay so we have this 3 things.

(Refer Slide Time: 21:34)



So we will start of a case where I have only k specific hypothesis in my hypotheses class. I am only searching thorough a space of k. k can be very larger, I am not telling how small or large k is but I am only saying, I have only k hypothesis in my class and I am going to search through this. It makes this slightly easier for us to developed some intution and the we can go on and talk about the infinite h okay.

So we want to look at how $\hat{\varepsilon}(h) \sim \varepsilon(h)$. For some hypothesis h how does $\hat{\varepsilon}(h) \sim \varepsilon(h)$.

I am going to fix some $hi \in H$.

The Bernoulli random variables that wanted here as we mentioned earlier are going to be defined by so random variable z is if $h_i(x)$ is not equal to y. It will be 1, it will be 0 otherwise.

So whenever hi makes a error then this random variable will be 1 whenever hi does not make an error the random variable will be 0.

So we can go head and write, write zj for each xj as xi xj not equal to yj so if you remember we always make the assumption that the training data was given to us in an IID fashion, independent

identically distributed fashion from very beginning we have been saying that training data is IID each sample was taken independently of one another and we used this fact even in the hypothesis testing case again we will use the fact here and that allows us to apply the Hoeffding bounds there allows us to apply the Hoeffding inequality.

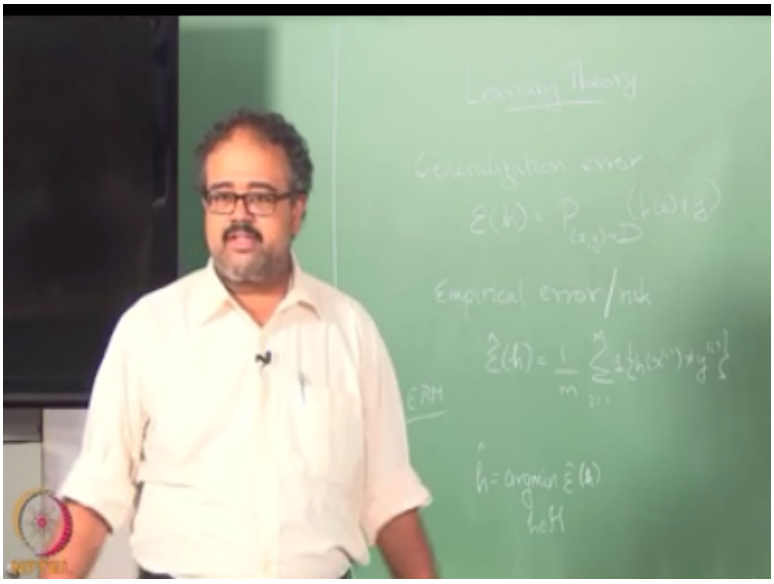
So what is that we have so I have if I have m training points like we said we assume there are m training points, therefore I have this m independent identically distributed random variables okay where is the distribution that I am drawing this from, from the true distribution of the data so I am assuming that all the points are coming from the true distribution of the data, all the training data points have come from the true distribution of the data so that is something which is very important.

These are the two main assumptions that we are making here what is the first assumption, IID the second assumption the training data comes from the true distribution so that is the distribution according to which I want to evaluate the generalization error so these are the two assumptions you make , of course we can always relax this assumptions in fact quite of we need to relax this assumptions because in real life we will not be able to satisfy either of the assumptions .

But this is good enough to give a some kind of a intuition as to how things will work and then we can worry about relax in this assumptions later on.

So ε is already define there, so all I have done is take whatever expression there was there in the sum there and define that as zj and I get this. so already we know that so what is ε(h) that is the true mean. What is $\hat{\varepsilon}$? so it is estimated by taking m samples. So you can directly, go apply some Hoeffding bounds.

(Refer Slide Time: 27:38)

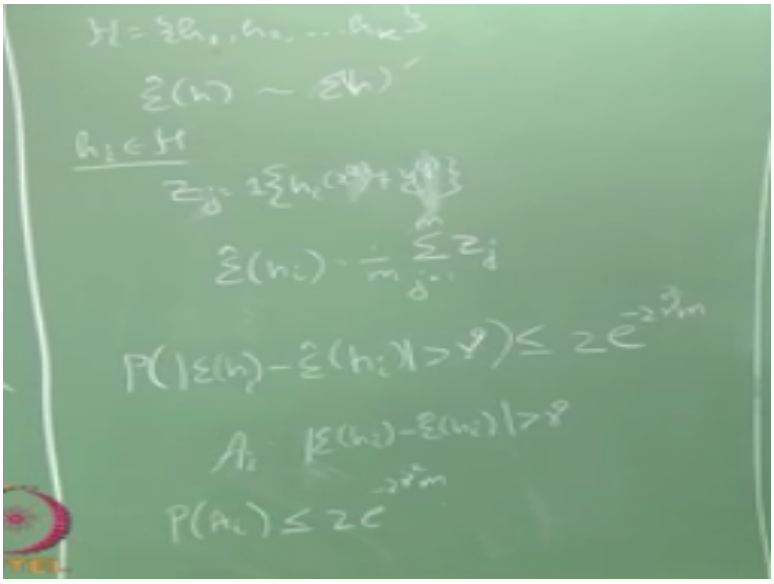


So if you have a very large class then you have to be very careful about minimizing the empirical risk, because you have you run the risk about fitting it , so yeah so essentially what you are trying to do in many of those cases is try to get a better estimate of the error on the better estimate of the you know generalization error actually, so when you are trying to do the validation .

So essentially trying to get a better estimate of the generalization error directly, so with all of these things or essentially to give you some notion of what is it, some notion of the complexity of the problem that you are trying to solve okay. This just gives you some notes of the complexity of the problem that you are trying to solve and we will see that in the minute you will see that .

So in fact one way to avoid over fitting in neural networks is to have a very, very, very large training set if you have lots of ways and we need to have a very, very large training set that will kind of all out of this just give me a second to explain this.

(Refer Slide Time: 28:48)



So what we are now shown here in the above image is that for a single hi from the hypothesis class if I have a large m then the error will be the probability of having a large error will be small, so or even say I will be pretty close even for small γ suppose I want to know what is a probability that this greater than ε greater than ε^ is the different between ε, ε^ greater than let say 0.01 so that is what I want to know so my ε^ should be within 0.01 of ε okay then what I will plug in here is 0.01.

Right so $e^{-0.01}$ so essentially getting to 1. Essentially getting to this since what I am saying is the probability of the error greater than 0.001 is less 1 then really tell me anything is just less than 1 at a I mean I know that, but then if my m is very, very large suppose I want this to be 0.01 this is 0.01^2 then I will let say m is a million so then what will happen is I will have e^{-20} or something, so that is a small number.

So if I say have a lot of samples then the probability of ε^ being 0.01 close to ε will be high .

what this states is the probability that it will be 0.01 away from ε is low , so the converse is if it is the probability that it will be 0.01 close to ε, it will be high so that is essentially the result that we have what we have shown here using the hoeffding bound so I will show the proof of the hoeffding bounds it is not trivial so interested you can look it up this is not very hard okay it is just have to work it out that is all.

So but once you accept that on faith though you have the result but unfortunately this holds only for one particular hi that is not very interesting I am essentially what I have shown is you can give me like a 10, 000 different hypothesis and I can show you in for one of those hypothesis, if I have a lot of samples then it will be close.

What I really want to show is okay if you give me 10, 000 samples are give me a 1 million samples for every hypothesis in this hypothesis class, I will be close so every hypothesis is in hypothesis class I will be close so how where you going to do that ? use the union bound so what I will say is I will define my event Ai is I need those A1 to Ak , I will define the event Ai as so define this event A_i as ε and ε^ being more than γ away so now this becomes probability of Ai.

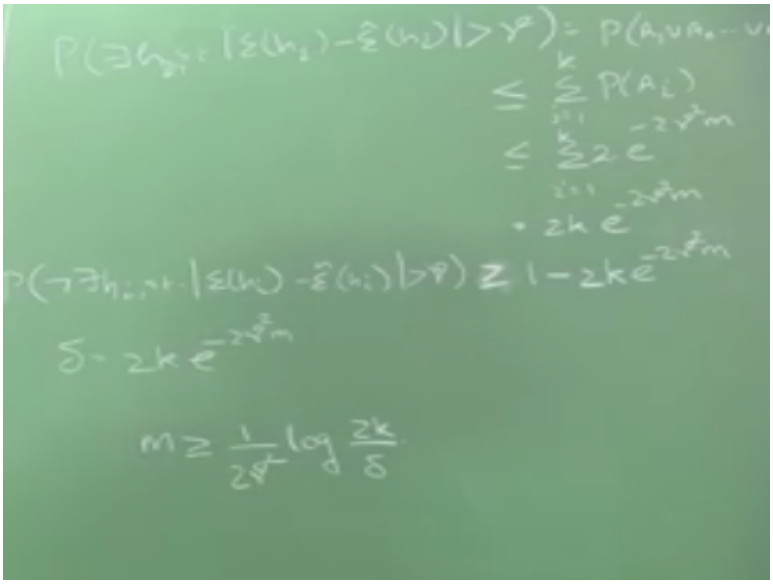
$$\mathbf{A_i : | \varepsilon(h_i) - \varepsilon\hat{}(h_i)| > \gamma}$$

$$\mathbf{P(A_i) \leq 2e^{-2\gamma 2m}}$$

Right now this is essentially becomes probability of Ai.

So now I do union bound so what is the union of A1 to Ak what does it mean at least one of them giving a higher error , so essentially this reduces to.

(Refer Slide Time: 33:43)



Where exist h such that it is that this is some hi okay at least one h so probability of at least one h for which there exist at least one h for which the error is large so this is essentially equal to the probability that there will be at least one error there may at least one hi which has large error is bounded by $\mathbf{2e^{-2\gamma^2 m}}$.

For one of them it is just $\mathbf{e^{-2\gamma^2 m}}$. For k of this it us just multiply by k this is essentially what we get from union bound.

So now what I need I need the probability that, that does not exist any classifier that makes a large error if I give it m samples okay that does not exist any classified that gives a large error so what will that be 1 – this similarly 1 – that. This is simple algebra here so we got this and so this kind of a result which holds for all H in the complexity class where mean in the hypothesis class we are taken if because this now this results holds for all H and capital H these are called uniform convergence results, . So this is the uniform result.

Because it holds for everything in the this is more like a single result this is for a specific hypothesis well this result that we are giving that $(1 - 2k)e^{-2\gamma^2 m}$.
That bound is for all hypothesis in this hypothesis class so this is called a uniform convergence result okay, when somebody says uniformly convergent that means set it works for all classifiers it works for all classifier, okay. So far note that I am not actually talked about finding the classifier, .

So what is ε, ε given a classifier what is the error in the classifier will be making in the overall population and ε^ is the error that it makes on the training data so I am just comparing the two I am not actually talked about finding the classifier so what we should not be looking at is, we should be looking at comparing okay I will come to that in a minute.

If you remember I said there are like three quantities in the beginning. So now I am interested in solving for m I want to know how much how many samples I have I want I can draw before I can give a certain guarantee on okay I can give a certain guarantee on the performance what I mean by performance here, whether my empirical error is closed to the generalization error I am not talking about the best generalization error when I am talking about performance here I am talking about whether my estimated error is closed to the true error, okay.

So how many samples should I draw before I can give some guarantees on the performance of the estimator okay. So I want to solve for m and I need to fix γ as well as the error probability as well as this guy so, k is fixed for me I give you the hypotheses class as soon as you give me the hypothesis class K is fixed for that so I am going to says that the probability should be utmost this quantity should be utmost δ like some δ I will give you the δ so this quantity I will fix so the probability should be 1 - δ.
and I will give you the γ also solve for m . will be , now I can give you γ also so I have given some value I will give you some number for δ.

I will say δ should be 10% so I will say δ should be 0.1 I will give you the δ I will say δ should be 0.1 .

I will give you the γ also okay, now solve for it. So what does equal to 0.1 mean that 90% of the time this event should be true okay so I will give you a γ I will give you a δ you find m for me. What this tells you is you chose a small hypothesis set to ensure uniform convergence.

So that it will tell you what the true error is it does not tell you anything about how good the true error is okay. So that is so that is what I kept re-iterating to say what performance means in this case, performance is not minimizing the error. performance is minimizing the error in predicting the error okay this is kind of a circular but this is trying to minimize error in predicting the error, okay that is all.Or in measuring the error.

$$m \leq \frac{1}{2\gamma^2} \log \frac{2k}{\delta}$$

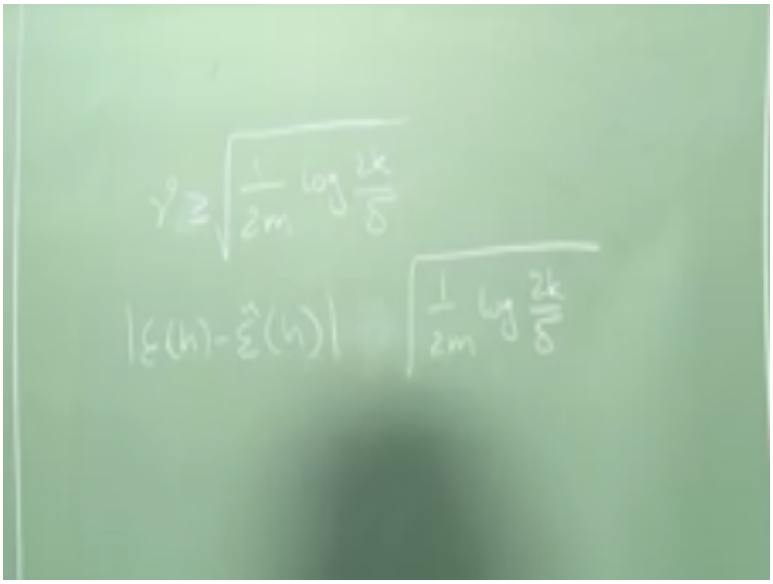
So this is called the sample complexity this tells you how many samples you have to draw so that what, so you are all your classifiers are within γ of the true classifier , the probability of that happening is at least 1- δ, . So this kind of formulation are called PAC formulations, P A C so you know what is PAC is?

You know what PAC is ? probably approximately correct , so that probably part comes from that, . The approximately part comes from that so I am not telling you it is correct okay it is approximately part comes from that so I am not telling you it is correct okay it is approximately it is within γ of the answer but is it always within γ of the answer no, no it is with high probability it is within γ of the answer, .

So it is probably approximately correct but in many cases this is the best I can tell you because there is so much variation in the samples that you are drawing and the problem itself has inherently has noise in it say there is only so much I can do in terms of predicting it correctly.

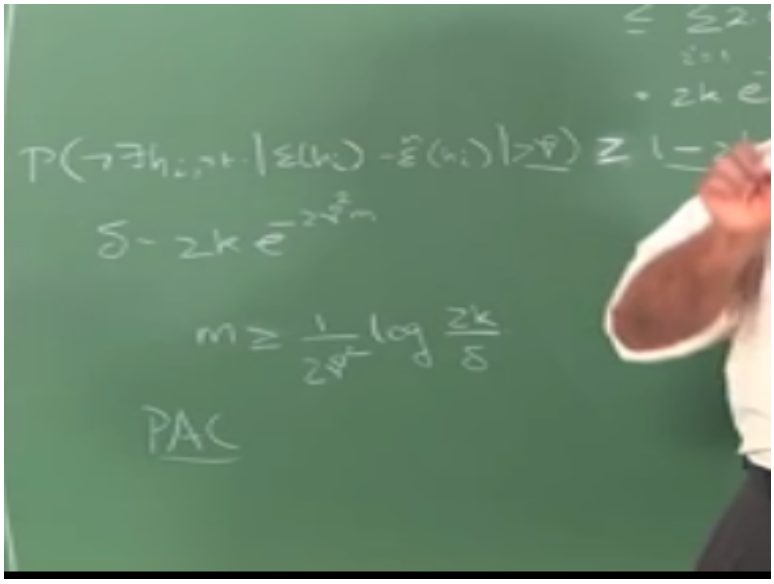
So the another thing which I want to look at is I give you m and δ okay. Can you solve for γ? Right I fix m and δ solve for γ what you get fairly straight forward ?

(Refer Slide Time: 44:50)



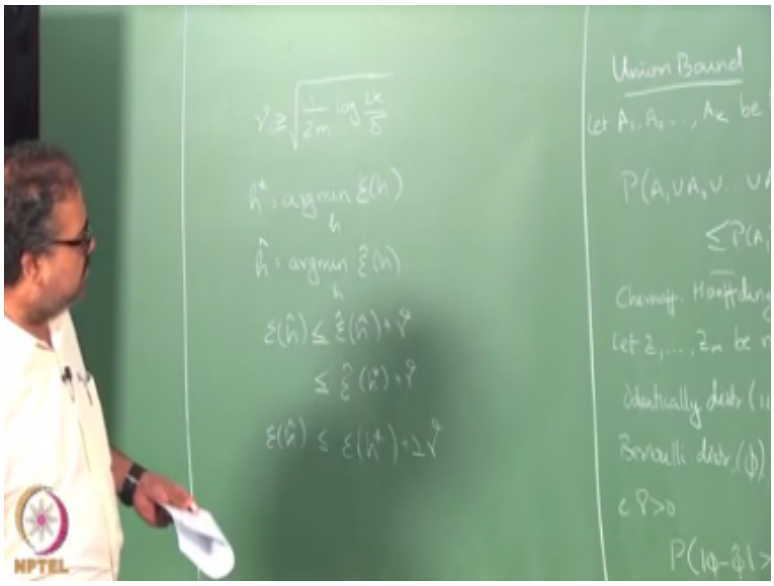
Okay likewise if you want to solve for δ you can try these things but this is fairly easy system so what is γ really it is the error in the prediction of the error .s so it is ε - ε^ is γ .

(Refer Slide Time: 46:50)



So my γ yeah so should be greater than or equal to here , γ should be greater than or equal to so my γ should be at least yeah, γ can be at most this small , γ can be at most this small but it can be greater so then I can give you the guarantee so that is essentially what we are looking at here , so this is not needed.

(Refer Slide Time: 48:50)



$$h^* = \text{argmin}\ \varepsilon(h)$$

$$\hat{h} = \text{argmin}\ \hat{\varepsilon}(h)$$

$$\varepsilon(\hat{h}) \leq \hat{\varepsilon}(\hat{h}) + \gamma$$

So we will define h* to be, h* is the true, the truly the best classifier that we have , in hypothesis class h , we will define h hat to be that classifier you will pick by doing empirical risk minimization , ε^ is your classifier you pick by doing empirical risk minimization okay, so knowing whatever we know can be write things , so $\hat{\varepsilon}$, $\varepsilon(\hat{h})$ is less than or equal to $\hat{\varepsilon}(\hat{h})$ + γ. why is this true because of whatever we have shown all the value , so I am saying that , I have taken enough samples M so I can say with some probability this event will hold, .

Because of my uniform convergence with some probability that some probability 1-δ this event will hold , because this will be γ close to the error , the $\hat{\varepsilon}$ will be γ close to ε.

$$\varepsilon(\hat{h}) \leq \hat{\varepsilon}\ (h^*) + \gamma$$

$$\varepsilon(\hat{h}) \leq \hat{\varepsilon}\ (h^*) + 2\gamma$$

So that means that h* should either have a higher error than $\hat{h}$ or at least equal error. It cannot be a better error than $\hat{h}$ because we have better error than $\hat{h}$ then $\hat{h}$ would not have been chosen okay, does it make sense , this I am using by the visual of the fact that I did minimization here , and therefore ε^ h* should be worse than ε^ h hat.

Using uniform convergence again I can peel out one more γ from there , did it make sense what I have done here is it is exactly I went from here to here , from here to here how did I write this γ because of uniform convergence , similarly what I did was I took ε^ h* and I said that it will be within γ of ε h* so then I add up the γ here so I get 2 γ , so all of this will hold with some probability but I am assuming that I have taken enough samples and I have converged to whatever degree of accuracy I need. I have converged to some probability 1 - $\delta\delta$. I have converged here.

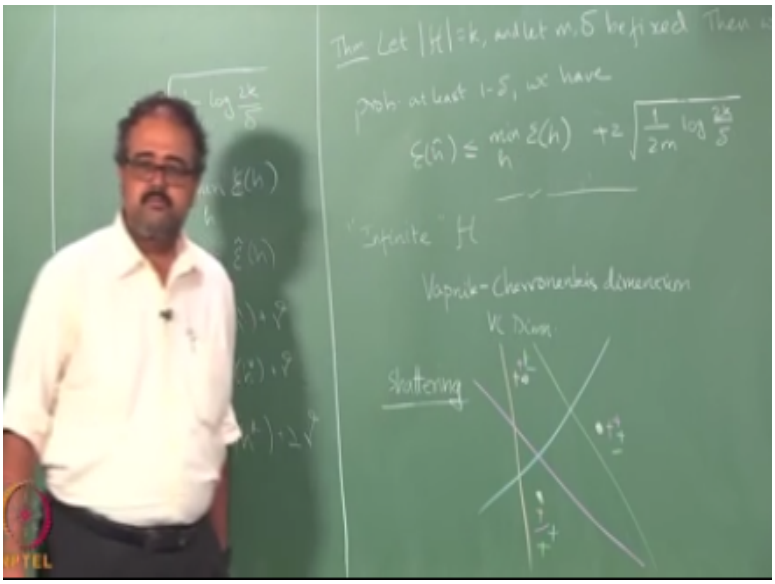
So once I have converged so this will hold so essentially what I have is that so what does is mean so the true error of the classifier I produce by doing empirical risk minimization is within 2 γ of the true error of the optimal classifier okay, does it make sense the true error of the classifier I produced by doing empirical risk minimization of which is $\hat{h}$ is within 2 γ of the true error of the classifier that is produced by minimizing the true error is essentially the optimal classifier, .

So essentially this gives you the guarantee that if I do empirical risk minimization , taking that many samples , then I will be within 2 γ of the true classifier anything else I need that, with probability 1 - δ.

so if I take at least this many sample M , so the classifier that I find by doing empirical risk minimization will it be within 2 γ so our γ is say number you plug in here , within 2 γ of the optimal classifier at probability 1 - δ , so that is essentially the result that we can have, .

So let us put this together and write small theorem.

(Refer Slide Time: 54:32)



$$\varepsilon(\hat{h}) \leq \min_h \varepsilon(h) + 2\sqrt{\frac{1}{2m}\log\frac{2k}{\delta}}$$

Right, so essentially I have written our γ from wherever we solve for γ, I wrote the γ expression down there , and we get this okay, . But what is ε (h*) , so ε(h*) is actually minimum over h of ε of h.

so this is essentially the best possible classifier that you can build okay. So now we have this result we can think about, if my hypothesis class is small , essentially what I will be doing is I will be searching over small number of things to find the lowest error so it is likely that my solution the best solution I can find in this class itself will be bad, if my class is small.

But if my class is small the second term is small , if the number of thing if the k is small the second term is small, but my k is large , then the likelihood of me finding a small error solution is high but the number of samples I will need will also become larger. So this is one form of bias variance trade off. So if the hypothesis class is small that means that regardless of how much data you give me I will always be making some error because I have only a few hypothesis that I can search through.

So that is like a bias you know it is like doing linear regression kind of thing. and this is variance because hypothesis class is very large then I will need a lot of samples for me to estimate the error properly, so that is the variance path so this is like one version of the bias variance trade off that comes in so that is the reason you need large hypothesis class so that you can be sure that you contain you have the solution in there.

But if you know exactly what is the solution we are looking for then you are better of using a much smaller hypothesis class okay, so that is essentially the to take away message here.

So we already know what is the sample complexity you need for this result hold for a specific γ , so here when the γ is given by this expression but if I give you a specific γ then say something like okay, with some probability 0.1, I want to be at least 0.1, close to the true answer , the probability 0.1 I want to be at least 0.1 close to the true answer.

So what is the 0.1 close here it is 2 γ , so it is 0.05 , γ should be 0.05 , so I will plug in 0.05 and 0.1 here , and I know what k is because of the hypothesis class I have chosen I can always find out what the sample complexity m is, so given a and a γ, I can find the m, okay. So this kind of an analysis, this kind of sample complexity is sometimes called is usually called ε where because people usually use ε as a symbol for γ.

But in this case it will be γδ sample complexity or γ δ PAC analysis.
Because I fix the γ I fix the and I ask you for the sample complexity , so this sometimes called γ δ PAC, okay. So this is assuming you have a finite hypothesis class what do you do in the case of a infinite H ? any thoughts about how it extent this analysis an infinite H?

So in a practical setting I will be typically implementing all my machine learning algorithm in a digital computer this is kind of a cheating argument but it is fine so I will be implementing these in a digital computer and digital even though they are implementing in infinite class of things they are limited by their by the numerical procession .

So let us say I use 64 bits to the present floating point numbers then only finitely number of finitely many classifiers I can represent with 64 bits . So the problem is it is a large finite number but I can still go back and if you look at that number I have there if you look at the m that I need you can think of m is being actually ignore a whole bunch of things here but m is order of $(1/\gamma^2)\log(k/\delta)$.

So you can always say that okay regardless of how large my hypothesis class becomes this is going to be log of that order of log of that so that will be a significant reduction. Unfortunately the hypothesis class becomes exponential suppose I have d I have d numbers I need to specify one classifier and have 64 bits , so how many hypothesis do I have? So how many bits do I need for representing one classifier? 64 times d so how many do you have and then how many classifiers I could have 2 power 64 times d, that is K.

I plug in K here then it becomes D it becomes order of D. So if I have a 1000 parameters then I need about order of 1000 samples not bad. 1 by γ, log all I mean it depends on what you except out of it you chose a large γ large I am just joking , yeah of course you always have those things the γ, and other things actually play role there but if you think about it I mean I have d parameters I mean to need to get at least order of D samples to solve the problem okay, how do you think it can do it less than order of d if I am do it less than order of D? Then many of my parameters are all tie together they are redundant.

So this the power of the big O notation you can hide a lot of things under the big o umbrella because I hiding all your 1 by γ square and log of 1 by you are hiding under the big O umbrella but it is not a bad thing okay so this is 1 way of thinking about it it is not the greatest way of thinking about it but this is one way of thinking about it in fact people use a rule of them

suppose you are training a neural network which has a say 10000 weights they use the typical rule of them they uses you need at least 10x the number of x, so if you have 10,000 weights you need 1,00,000 data points.

At least in fact this is a very useful rule of thumb if you are only using feed forward neural networks so remember this if they count the number weight you have and ten times that how many data points you need at least for you to give anything useful. But then there is a better way of doing it . call the Vapnik - Chervonenkis dimension otherwise known as the VC dimension .

So given a hypothesis class we can define the VC dimension of then hypothesis class I will finish in a few more minutes okay. So before I define a VC dimension I need to introduce the notion of shattering , so give some set of points okay let us say some set S, I give some point x1 to xk let us say a hypothesis class H is set to shatter the set S if for every labeling you can give for that set S, there is some element in the hypothesis class which actually separates the classes a binary classes okay.

So every possible binary labeling that give on the set okay I have a hypothesis in my hypothesis class that separates it from one class from the other okay is it clear, so let me draw a picture that will make it clear let us say that, so that is my set s okay and my hypothesis classes all straight lines okay. So think of all possible labeling I can do for this . I can basically set on say everything to one side of the line is + everything to other side of a line is - great.

So now what we do everything to one side is + everything to other side is - so likewise I can keep going I keep going as long as there are different number of colors here and then guys can kind of intute . So give the set of three points you can just see that so if I flip the + and - it is exactly the same so you have to only worry about the unique things. so if the flip the + and - it is just the same.

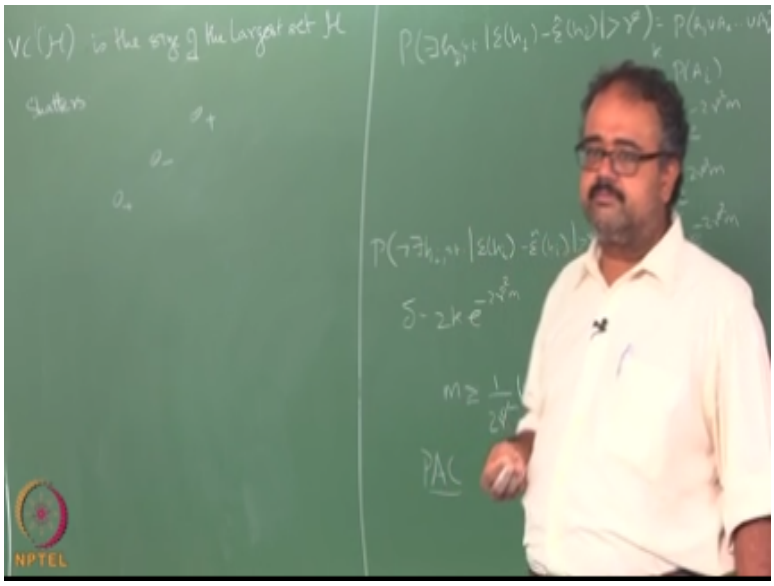
So if I make this + and these two - the same thing will work . So is there anything else that we need to consider? So + + -, + + +, + + - that should be a - , that will work anything else so I have to consider 2 pluses and one - two - and one + is just flip of it okay and three pluses I have

considered and three - is just the flip of three + okay, so anything that we can to consider? That is it ; I will leave anything but anyway even if I left out something you can make it up , so you can easily see that.

So straight lines so hypothesis class of straight lines shatters three points in space , what about four points? Yeah we talk about single straight lines, so what about four points? Is there any configuration of four points which can be shattered by single straight lines, no configuration of four points that can be shattered by single straight lines that then automatically applies a five six seven eight nothing .

So the VC dimension of a hypothesis class H okay is the size of the largest set that the hypothesis class shatters.

(Refer Slide Time: 01:10:50)



Note that this is the size of the largest set high H shatters that does not means that H has to shatter all point of that size even in the three case co-linear points I cannot shatter if may three points like this and I label this + - + I know straight line that can separate this okay but there is

some configuration of three points in fact lot of configuration of three points which I can shatter and therefore the VC dimension of straight lines is three .

What about VC dimension of pairs of straight lines parallel lines not arbitrary pairs. Let us say I will give you parallel lines what is VC dimensions of parallel lines, 2 parallel lines, they have to be parallel. However you want you can look at it, but it has to be 2 parallel lines, it can be 4 for sure. Can you do 4? You can always keep one of them as redundant straight line you can do as much as you can, and then you get you get straight lines. As everything else can do by 1 line, only the Xor case + + and - - you cannot scatter with the single line .

Remember perceptron, so it is exactly that, so for that you need parallel lines. Everything else you have 1 lines there and you keep next parallel line at infinity you are fine. What about 5 points? Now you can see why we can ask all kind of interesting questions . So I can give you this kind, next I can give say okay, instead of looking at parallel lines, just look at quadrilateral . So if it inside the quadrilateral it is class 1, class positive is outside the quadrilateral its - okay. Now what is the VC dimension, in fact there are uniform conversions VC dimension as well.

So I will have the TAs put up this write up online and there of course other material you can find out online. I am not going to do the proof because it is pretty complicated derivation , but then the nice thing about this is, at the end of the day just like we have staple complexity in the number of parameters . You can show even with the Vc dimension that it is polynomial in the Vc dimension . So if the data as the hypothesis finite Vc dimension the uniform conversion, require you to have order of the Vc dimension of samples.

And typically it turns out that for most kind of classifiers, most kind of reasonable classifiers that are out there, Vc dimension will be of the order of the number of parameters in the classifier. So if we think about it straight lines. How many parameters are there in this case? 2 or 3? 2 slope and intercept and slope that is all you have. 2 parameters and the order, in the Vc dimension is 3 close. This will be close to the parameters you have , so and that is why I said 10 time the f weights in the neural network rights all of these things and you can get that kind of rough intuition.

So I will stop here so if you have any questions feel free to fire away. So you could do pack line regression , but Dc dimension is different for classification. Yeah you can coarsen the regression problem little bit then try to do dimensional classification and PAC you can do for regression, essentially you are trying to look at. We defined a very specific variable random and did it, so you could define any random variable. Whatever the distribution is there, it is the amazing thing about the Chernoff Hoeffding bounds .

So the result holds on the expectation, the parameter μ whatever is the expectation of the distribution, so the empirically evaluated expectation will be the close to true expectation that is the result we have. With that you can change your random variable definition and you can get something appropriate for the regression as well. What does it mean? When you change the parameters it is the different classifiers that what I am concern, Do you mean same family of classifier? It depends on how you define hypothesis class .

So in at the end of the day what I am really interested in this is? What is the decision rule the hypothesis class is entailing for me. So I can say that I am going to define hypothesis class is the mix of decision tress and something it is up to you but typically you define hypothesis class as a single family which is differentiated with the parameterization. But we do not actually take a call on that, all I am telling you is that okay given k hypothesis , how are you are going to find out the.

So what I am finally, at the end of I am interested in the decision rule, okay given a data point what does it assign it to you? You derive the decision rule by the means of using a logistic regression or whether you derive the rule by means of using I don't know LDA it does not matter. For this kind of complexion, how you got to it that does not matter for me. Only in this context otherwise it matters, with the context of the complex analysis it does not matter.

Which is is our finite procreation arithmetic is little disappointing? Because there I chose D parameters , if you have got it from some other I could re paradise thing and I can increase the number of parameters for representing the same decision surface. I want to define the straight

line but I can actually increase the number of parameters that I am going to use to define the straight line, I will be saying that where ever I have ax square, I will have - ax square . so now I will have two more additional parameters for me to define one should a and other should be - a.

And it will take out the difference of x^2 so it will still be a straight line but I have increased the number of parameters and going back to our finite procedure, arithmetic argument I also have blown up my complexity, which is casually not correct, all I am interested is the decision rule, that is why the finite thing is little unsatisfied. But here we did not tell you how you could represent the line. So Vc dimension definition does not require you to know to represent the line. At the end of it you know that it is the hypothesis that I want to represent.

But the most compact way of representing the hypothesis parameters I would need. You do not have to worry about how we get to it. Anything else, I can chose any classifiers that I want from my set , foe shattering the data points, so any effect it will 1 corresponding to the, I mean what is the most powerful, so other one will shatter that would be Vc dimension. So Vc dimension if it can shatter of any arbitrary size VC dimension will be infinite. So if Vc dimension of the classifier is infinite none of the analysis will work, all the Vc dimension analysis works assuming that Vc dimension of the classifier is finite.

I should point out that most of the classifiers we looked at do empirical minimization except? Anyone know? SVM, is called structural risk minimization because they have an additional constrained that is there apart from the empirical they also try to minimize the solution size. They try to minimize the norm of the weight factor so that actually gives rest to a different kind of minimization. So it does not do empirical, they called structural risk minimization. So you know who came up with SVM? So vapnik.

So he said came up with structural risk minimization this is the best you can do, so we need to have a different way of doing the minimization, then he motivated and then he said and came up with empirical structural risk minimization you can do. We will go and do something else and he can do something else structural risk minimization and then he derived SVM with them, if you

read original presentation of SVM it will not look anything like how we presents SVM now days. If you remember we started off with perceptron and then we went from there.

He starts from the theoretical things, he said okay if you do empirical risk minimization it is the best you can do what else I can do and then I can improve on it and then he came up with structural risk minimization okay and then he derived all those systems okay.

IIT Madras Production

Funded by
Department of Higher Education
Ministry of Human Resource Development
Government of India

www.nptel.ac.in

NPTEL

NPTEL ONLINE CERTIFICATION COURSE

Introduction to Machine Learning

Lecture-81
Frequent Itemset Mining

Prof: Balaraman Ravindran
Computer Science and Engineering
Indian Institute of Technology Madras

Okay, anyway so remember we actually spoke about frequent pattern mining very briefly at the very first lecture, where I was introducing different machine learning tasks to you.
So this is a form of unsupervised learning and the statisticians call this as, I told you that also, statisticians call this as bump hunting.

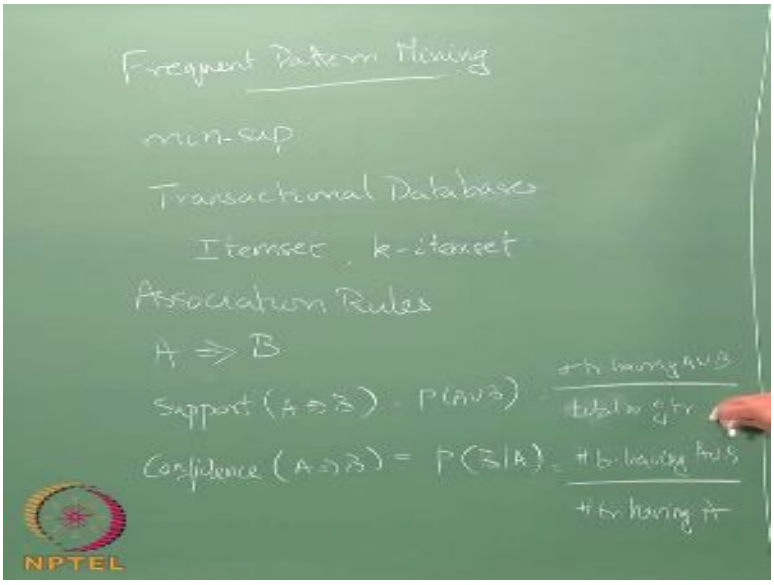
So you have a remarkably flat probability distribution, then there are small bumps somewhere, so what does it mean, slightly more frequent in the data than the rest right. So that is essentially what bump hunting means. So I have a very large pace, I have an exponentially large set of possible outcomes, and I am going to have an extremely low probability of seeing any one outcome. But there will be small bumps here and there which are places which are slightly more frequent than what I would see normally right.

So what, in the modern context right, so, something like Amazon's logs, if you look at who bought what on Amazon right. So if you think about it Amazon has millions and millions of transactions right. So say, suppose they have in a month, let us say they have like 10 million transactions on Amazon. If somebody bought say 10,000 copies of the same item, not somebody like some item was sold 10,000 times on Amazon right.
That is a very frequent occurrence, but think about what fraction of 10 million is 10,000.
1% ?, 0.1% ?, 0.01% ?, 0.1% right. That is why I call this bump hunting. It is remarkably flat right, so I have a huge inventory in my Amazon base, my Amazon catalog has a huge inventory

right, and I am looking for frequent items there which is like something sold 10,000 times is frequent for me.Even though the overall transaction is 10 million.

So this is essentially what I am talking about when I talk about frequent pattern mining. So you have to take the frequent path here with a pinch of salt. But, all the examples and illustrations I will give you, a frequent will be like 50% of the data or something. That is because I cannot draw 10 million things on the board right. But in reality when you actually use these kinds of things, the numbers will be or the fractions will be very different. Just keep that in mind.

(Refer Slide Time: 03:07)



So the frequent patterns are those that are above a certain, suppose above a certain minimum support that I am looking for in a, so what is the support of a pattern in a database ?. That is the support count. So support of an item is essentially the fraction of times it occurs. So take the support count, the number of times it occurs, divided by the total number of items, and it gives you the support of the item.

So I will call an item as being or all call a pattern as being frequent if it is above the minimum support. So minimum support is a parameter that I define, which is less than one, but it is a parameter that I define. So occasionally what people do is, they actually translate the min support also into a count. Essentially you take the fraction, and you multiplied it by the total number of items it gives you a count.

So see, sometimes easier to think of it as, I have a transaction of say, I have a total database of 10 possible transactions, I look for a min support of 2. So that is like 20%, but then you could think of it that way as well. So min support is essentially the minimum support level at which I will consider something as frequent. So classically, frequent pattern mining was applied to transactional databases. So where I have, a collection of transactions, a transaction essentially could be things like, these are the items you bought together, or these are the items you checked out together from a bookstore or library, or these are the items you borrowed from a library together, some kind of a transaction. So something went from A to B. So that is the usual, classically where they applied this, and as you are mentioning market basket analysis, is a place where they did this first.

So why is it called market basket analysis? So you go to a shop , you go to a supermarket, you buy something in a basket right, generally you bought a basket along with you, start putting things from the shop into the basket, then you come and get it checked out right. So everything that goes together into the basket, we call a single transaction. You might go, you might buy some cereal, you might buy some milk, or whatever you want to buy right, some vegetables, everything you put together and you bring it to the bill. So all those things that go together we call a transaction. So market basket analysis is essentially, analyzing what goes into your basket right.So this is the kind of things we will have. These transactions will essentially be defined over a universe of items, and each transaction will be thought of as a subset of these items or as the data mining people call it will be referred to as an itemset.

So what is an itemset ? It is a set of items, that is it. So instead of calling it a grammatically correct fashion, a set of items, for whatever reason, they introduced a new noun called itemset. It is a single word. They introduced a word called itemset and then they started calling it frequent itemset mining and so on, and so forth. But we typically be using the word itemset and we will also use a term K itemset. What do you think is the K itemset? Set of size K, set of items of size K. That is the K itemset.

And so, we also have something called association rules, that we talked about in the context of frequent pattern mining. So what is an association rule ? These are rules of the form $A \Rightarrow B$. What does it mean ? It means that, if you buy items in the itemset A, then you are likely to buy the items in the itemset B. So A and B are sets, they are individual items, and A and B are sets.

So if you buy things in the itemset A, so for example, if you buy the usual thing, offer is if you buy milk and bread you are likely to buy eggs or something. So you are basically going out shopping for breakfast items. So you buy milk and bread and then you also buy eggs.

Now a very famous or infamous example that people had is, if you go out to buy beer you also buy diapers. So why do you think that is ? causal effects ?. You are not kids, you are supposed to know this. Do not laugh at this. Yeah, some of, yeah the masters and PhD students can laugh. I am just kidding. So yeah, so why do you think that was the case?. Any theories?. Come on you should be knowing, give me some theories. People have been giving you horror stories about having kids, it is not that bad. And so there is another, then people did some analysis and they actually found all this is like this, the spike was happening on Sundays right. So in the US Sunday is football day. Every Sunday they have football playing. These guys are actually buying beer to drink during the football game. So at the same time, they also pick up diapers, because they do not want to be disturbed by the baby during the game or something right. So they probably slap a couple of diapers on the baby, let us say okay, do not call me when the game is going on. So that is basically what was happening.

So there is a larger point to this. So it is not enough for you to do association rule mining. Now some of the associations you discover just from the data might not immediately have any meaning to you.

For example, you cannot say buy two whatever, two crates of beer and get a diaper free. So you cannot use it for promoting sales or also it will look really weird if you start stacking diapers next to the beer cans in the shop.

I always use an example to illustrate the fact that statistics is all fine, but you need something more than statistics in order to get any useful intelligence out of data. So you need to think of other ways of doing it, but we are not going to do that.

So how do I know this $A \Rightarrow B$ is useful, ignore the discussion we had. Normally how will I say whether it is useful or not ? So I have different measures by which I can measure the usefulness of a rule. So two most popular ones or called support and confidence.

So what do you think is support ? How many times A and B have occurred together. So this is essentially the $p(A \cup B)$.

$$Support(A \Rightarrow B) = p(A \cup B)$$

So this is essentially, the number of transactions having A union B divided by the total number of transactions

$$Support(A \Rightarrow B) = p(A \cup B) = \frac{Number\ of\ transaction\ having\ A \cup B}{Total\ number\ of\ transactions}$$

$A \cup B$ is a set union. How many times A and B have occurred together.

And the second thing that we look at is okay. Yeah, yeah a set union, A is a set, B is a set, so when I do the union of that, all the elements in A and B, what is the probability that all the elements in A and B have occurred. This is how it is usually denoted right, so it is a set union.

So this had been literals, then I would have put A and B, what is the probability of A and B, but since this is a set, it's A∪B okay.

And confidence is essentially the p(B|A). I am saying A ⇒ B that means, then I have to figure out, okay how many times did B occur when A occurred. So the confidence of A ⇒ B is the p(B|A).

$$Confidence(A \Rightarrow B) = p(B|A)$$

This is the number of transactions having A union B divided by the number of transactions having A.

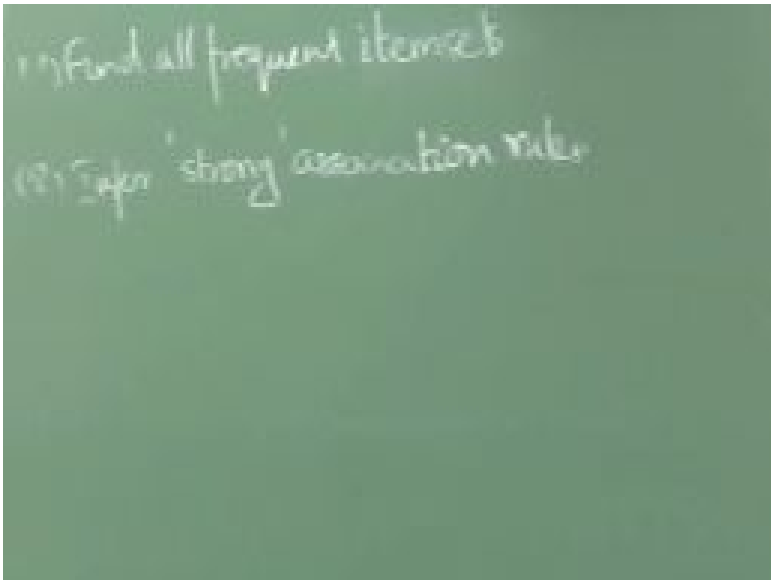
$$Confidence(A \Rightarrow B) = p(B|A) = \frac{Number\ of\ transaction\ having\ A \cup B}{Number\ of\ transactions\ of\ A}$$

So how do you find these association rules ?

You first do frequent pattern mining. Find all patterns that are frequent. Then you will find that A is frequent and some A∪B is also frequent. Then from that, I can start inferring these kinds of association rules.

Yeah you are right but it is still a hard problem. So that is what the rest of the class is going to be about, how will you do this efficiently.

(Refer Slide Time: 14:57)



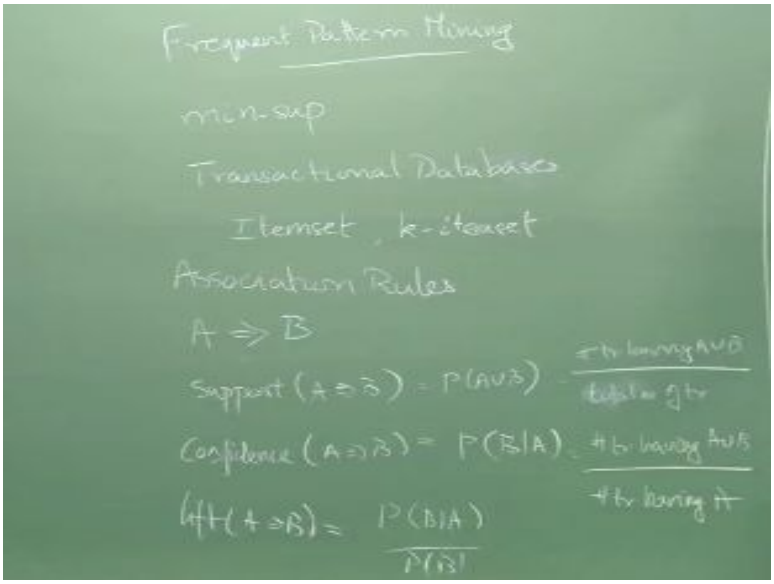
So usually how you do this is, the first step is, find all frequent patterns. Second step is, infer strong association rules. Let us get into the habit of calling patterns as itemsets. So all frequent itemsets or itemsets which have min support. All strong association rules will be those association rules which have min support and as well as a min confidence. Like minimum support and minimum confidence. I want both, right for a strong association rule.

But there is a caveat. Just having strong support and strong confidence alone is not enough. So you will also have to see what is the probability of B in isolation.
I look at probability of B given A, p(B|A) and I say it is 0.6 . That looks like a good association rule. But if I remove A, I just look at what is the probability of B, and if I say the probability of B is 0.75, so then what happens ? A implies a depression in B actually. I should not say that if A occurs then B will also occur. If A occurs then the chances of B occurring will go down.

So this is something that again a classical example is, when people are analyzing data from a store called blockbuster right, so blockbuster rents videos they also sell video games okay,
and they found out that if people rent videos from the shop, I am sorry, if people buy video games from the shop, they also rent videos. That rule had a confidence of 0.6. But then, if you do not buy a video or if just anybody who comes to the shop whether they buy a video or not, I am

sorry, buy the game or not, there is a probability 0.75 of them renting a video. So essentially if you go to the shop to buy a game, you are less likely to rent a video right, so you have to be careful about that right.

(Refer Slide Time: 17:52)



So there is another thing which people use which is essentially the ratio of the probability of B given A and, the probability of B.

$$\frac{p(B|A)}{p(B)}$$

So if this is greater than one, then knowing A is useful and if it is less than one, then knowing A is not useful.

$$\frac{p(B|A)}{p(B)} > 1 \text{ , knowing A is useful}$$

$$\frac{p(B|A)}{p(B)} < 1, \text{ knowing A is not useful}$$

This quantity is sometimes called lift.

$$Lift(A \Rightarrow B) = \frac{p(B|A)}{p(B)}$$

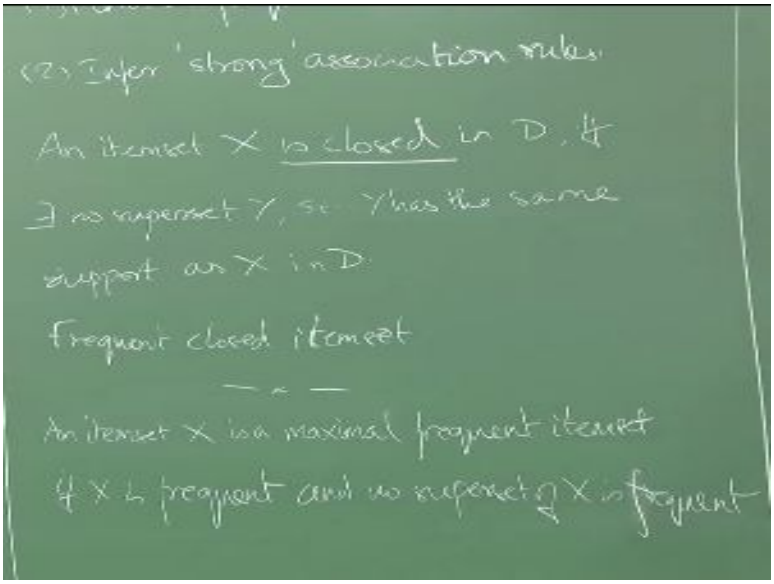
So lift also has another interpretation. Sometimes people take the difference between the two and that is also known as lift. Sometimes people take the ratio. I think nowadays ratio has kind of become the standard way of defining lift.

I should tell you that association rule mining is a very, very popular subfield of data mining. I have given you three different ways of measuring usefulness of rules and there are about a 100, and I do believe Nandan covers a good fraction of those hundred in his courses. So if you want to know more about it go to dooms right.

So that there are lots of different ways of measuring this but these three are pretty common. Support and confidence are the base right. And then people build a lot of things on top of support and confidence. So that is basically it.

I am not going to talk about association rules any more. The interesting problem as you could have rightly surmised by now is finding all frequent itemsets. So just a couple of other definitions here.

(Refer Slide Time: 19:57)



An itemset X is closed in a particular dataset D, if there is no superset of X, that has the same support as X in D. So it has to have a lesser support than X, then you call that a closed itemset.

What is a frequent closed itemset? A closed itemset whose frequency is higher than min support. So if I give you the counts of all the closed itemsets in my dataset, I am sorry, all the frequent closed itemsets in my dataset, you can recover the counts for all the frequent itemsets in the dataset.

If I give you the counts for all the frequent closed itemsets, you can recover the counts for all the itemset in the datasets, fairly straightforward, because when would the itemset be part of the closed itemset ? If a superset has a lesser count than it. But the superset could still be frequent. Let us say I have a frequency threshold of two and some itemset has a count of five
and add one more item to it has a count of only four. But still that is also frequent. It will be part of the closed frequent itemset as long as that is a superset of that, that has a smaller count.

If it doesn't, what does it mean ? If I say that A,B,C is a closed itemset and there are no two itemsets that are closed. What does it mean ? The count of A,B, the count of A,C, and the count of BC is the same as the count of A,B,C.

To get that, if we say that A,B,C is a frequent closed itemset, that means that, let's say it has a count of five, that means that A,B is also a frequent itemset, and A,C is also a frequent itemset and B,C is also a frequent itemset.

{A,B,C} - 5

And what are the counts of {A,B}, {A,C} and {B,C}? Five right.

So if I give you the frequency of all the frequent closed itemsets, then I can recover the frequency of all the frequent itemsets. The reason is called closed. This is sufficient for me to recover the entire data.

So typically if you are trying to come up with a new counting algorithm for itemsets, you have to make sure that you return the complete set of closed frequent itemsets right.

A itemset is a maximal frequent itemset, if X is frequent and nothing larger than X, no superset of X is frequent. So it is a severe case of closed. So the closed condition is, the superset should not have the same count, it should have a lesser count than the subset. But it could also be frequent. Here I am saying, not only should the superset have a lesser count, but the count should be so less that it is no longer frequent.

So the set of all maximal frequent itemsets will be smaller than the set of all closed frequent itemsets right or frequent closed itemsets. So that will be smaller. And is the maximal set sufficient for you to recover all the frequent itemsets ? No ? Not necessarily ?
Sure, sorry, we did not say that.

So I mean there has been equal frequency, yeah but I cannot give you the frequency of those. But I can say which are which are all frequent. But I cannot give you the frequency of those subsets. While in this case I can give you the frequency of the subsets, because they will be the same. So I can still recover something about the frequent itemsets, but I will not be able to tell you what is the frequency.

IIT Madras Production

Funded by
Department of Higher Education
Ministry of Human Resource Development
Government of India

www.nptel.ac.in

NPTEL

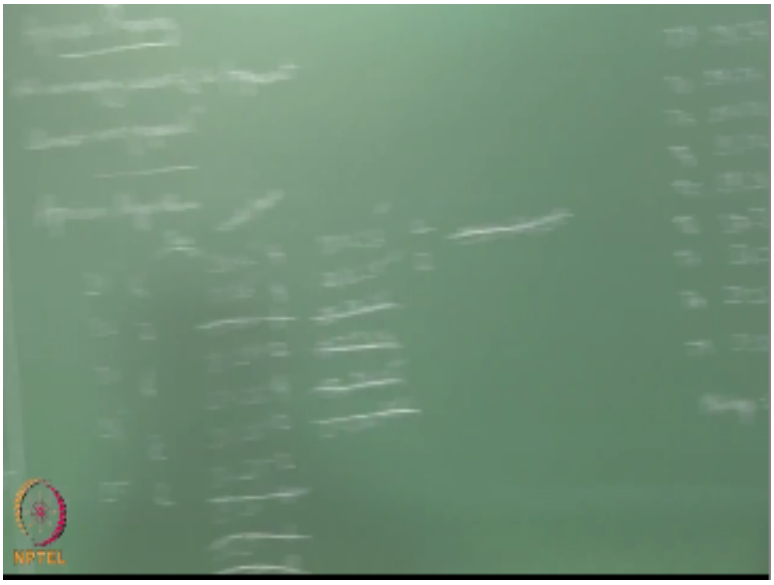
NPTEL ONLINE CERTIFICATION COURSE

Introduction to Machine Learning

Lecture-82
The Apriori Property

Prof.Balaram Ravichandran
Computer Sicence and Engineering
Indian Institute of Technology Madras

(Refer Slide Time: 00:19)



Great, so how do you find the frequent itemsets. So we will make use of something called the apriori property.

What is the apriori property ?

All non-empty subsets of a frequent itemset are frequent.

Very simple, so you use this idea for pruning the candidates that you will generate while you want to count. Most frequent pattern mining algorithms go like this, so you start off with a database of transactions and then you generate candidate item sets. These are all the possible

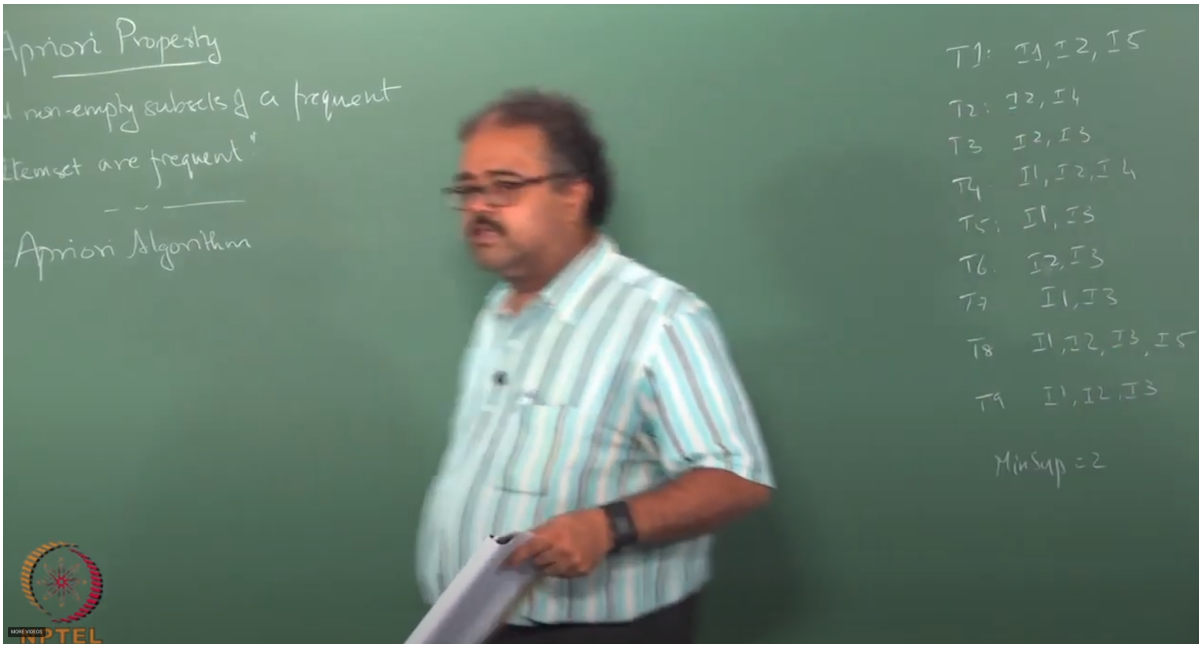
item sets that could be frequent, then you go count them and then based on the counting you can prune away some of those.

If you are doing this blindly, suppose I give you a universe of five elements how many item sets can you generate. So I can generate two to the power of 5-1 candidates then I have to go count all the two to the power of 5-1 candidates I have to count . Five is fine what if its 2 to the power of 10 million – 1 .

So that is not going to work and you need some very strong way of pruning these things, in fact whatever I am going to talk to you about today will not work if you are Amazon you need even stronger ways of pruning things. There are other techniques for doing it so based on this apriori property, people propose something called the **apriori algorithm**. So apriori algorithm is not the only one that uses apriori property a lot of frequent pattern mining algorithms used the apriori property but there is a specific algorithm called the apriori algorithm .

So what I am going to do is I am going to talk to you about two different algorithms for doing this. I will just do this by illustration. Start by taking a database of transactions and then I will walk you through the steps for doing this counting.

So I just write down the database here and I am going to say that I require a min sup of two.



The apriori algorithm proceeds in passes. In pass one, I find out the frequency of all one item sets. Now what we do is I do any kind of pruning I want, so I throw away all the one item sets which have below the minimum support threshold. So I throw away I6, I7 all that .

I1 – 6
I2 – 7
I3 – 6
I4 – 2
I5 – 2

Now what I do is, for people who know databases, I do a self join and generate candidates, and for people who do not know cell join I basically extend all these patterns by one. All possible extensions . So why is join an interesting thing because join has been highly optimized by the database community. So if I just say go ahead and do a join, I can compute it much faster than doing a sequential scan of this and extend each pattern by one. I am assuming that everything is commutative. So once I do I1 I2, I do not have to do I2, I1 because they are all sets and these are all the candidates for the phase two.

I1 – 6		I1,I2
I2 – 7		I1,I3
I3 – 6		I1,I4
I4 – 2	JOIN	I1,I5
I5 – 2		I2,I3
		I2,I4
		I2,I5
		I3,I4
		I3,I5

So what I do now is I can do a pruning what is the pruning I will do. So remember I am going to use the apriori property. So this has to be frequent all that subsets have to be frequent. So what are the subsets of this well I1 and I2? It turns out that since I generated this join from one itemset table which are all frequent so all of the subsets will be frequent so in this case I do not have to do any pruning. I will just count.

So the count is as follows:

I1 – 6		I1,I2	4
I2 – 7		I1,I3	4
I3 – 6		I1,I4	1
I4 – 2	JOIN	I1,I5	2
I5 – 2		I2,I3	4
		I2,I4	2
		I2,I5	2
		I3,I4	1
		I3,I5	0

Ok these are the counts for this. Now I can do pruning. So anything that is not frequent I will throw out.

So what I have done here I have done a count and then I did a prune . Now what do I do again whatever is left I do a self join. Our for non CS people I extend it by one more provided I1 and I2, I mean so the first elements are common .

So for example I can do a I1 I2 I3 . So I cannot do I1 I2 I4. I cant extend I1 I2 by adding I4 because I do not have an I1 I4 . I need the first elements to be common when I do self join. So if I want to do an I1,I2,I4 , I need I1,I4 here. Since I don't have that I cannot do the I1,I2,I4 join.

Ok so this mean additional join gives me an additional pruning and taking advantage of my set property that the order does not matter so I am doing an additional pruning. So likewise I can do these 6 things. I1,I2,I3 , I1,I2,I5 , I1,I3,I5 , I2,I3,I4 , I2,I3,I5 , I2,I4,I5 , these are the six elements I will get after I do the join .

Now even before I count I can do some pruning. How? So I look at this so can I prune the first two? No I cannot prune the first two, but I1,I3,I5 , I can prune. Why because the subset I3,I5 , it is not frequent. Even before I do the counting I can do the pruning. This is where I use the apriori property. What about I2,I3,I4 can I prune ? I3,I4 is gone, so I can prune that. What about I2,I3,I5 again I3,I5 is gone I can prune what about I2,I4,I5, I4,I5 is gone I can prune.

So all of this I prune even before I count. So now I only have two item sets of size 3. Two, three, item sets that I have to actually go and do the counting for.
Will there be a join after this ? Yes I could do a join after this and the joint will be I1,I2,I3,I5. First two will have to be common so I can do a join so after this will do a join which will be I1,I2,I3,I5

And will this be frequent ? Our friend I3,I5 comes to our rescue, so this will not be frequent so I am done. So what are all the frequent itemsets?
So what is the big drawback with apriori algorithm. You do generate a lot of candidates and then you keep pruning them but even here even though you pruned a lot of the candidates without counting but you did end up generating a lot of unnecessary candidates and then you have to go back and verify the apriori property for them and then prune them that is one drawback and the second drawback is in every phase you scan the data all over again and you do the counting. So when you counted this, you do not somehow save the information for generating this count as well.

So there are lot of newer frequent pattern mining algorithms that work on very large data sets which they try to do it with a single pass through the data. They keep ancillary data structures around okay they do a single pass through the data and they are able to count for all the frequency.

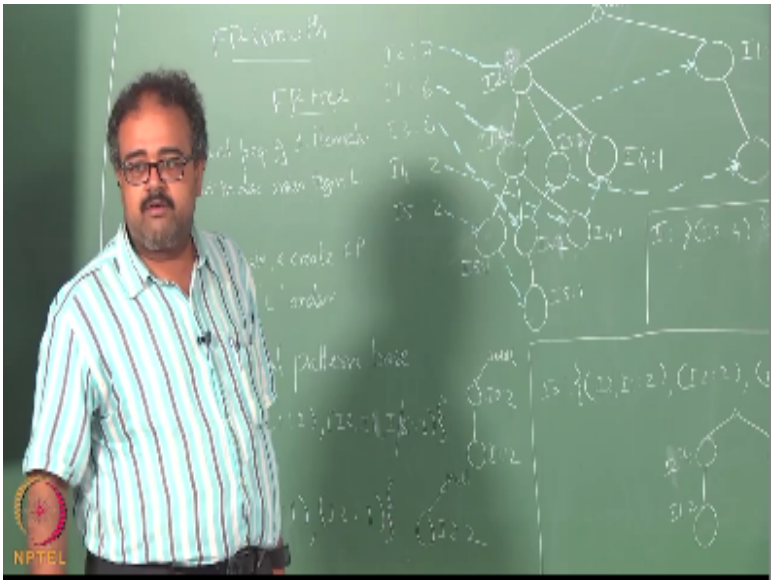
You would think that you would need at least two passes, one pass to generate the candidates and one pass to do the counting, but there are ways of doing it without actually doing that a single pass algorithms are available but I am going to talk to you about a two pass algorithms now, which is efficient. Everybody get the apriori algorithm ? It is fairly easy.

So let us do the two pass algorithm now.
FP growth. Any idea about what FP stands for ? Frequent pattern growth , or FP growth.

So it tries to avoid unnecessary candidate generation and minimize the number of passes you go over the data. How does this accomplish this ? They accomplish this by creating a data structure called the FP tree. So once you created the FP tree, you make many passes over the FP tree, but it is a somewhat of a compact tree. Not a balanced tree, more like a kind of a trie like data structure if people know that but you do not have to know that .

So you just go over the tree, it is much more compact representation than going over the table all over again. So you know once you construct the tree the tree can stay in memory and you essentially go over that tree again and again. So let us do this step by step. We will construct the FP tree first again then from there FP tree will start generating this.

(Refer Slide Time: 16:12)



So what we do, first phase is you go over the database once and then you count. What do you think you are going to count ? The frequency of all the one itemsets. That you have to do, there is no other go there. You count and you sort this by descending order. So let us call this order as some L. So the L is this ordering now what I have to do is go into each transaction and reorder the items by L .

I1 – 6
I2 – 7
I3 – 6
I4 – 2
I5 – 2

I can do this reordering as I do my second pass. In the first pass through the data I count the one itemsets. In second pass through the data I create the FP tree in the L order so L order is essentially each transaction.

T1	I2,I1,I5
T2	I2,I4
T3	I2,I3
T4	I2,I1,I4
T5	I1,I3
T6	I2,I3
T7	I1,I3
T8	I2,I1,I3,I5
T9	I1,I2,I3

Now forever transaction I create a path in the tree. So I will start off with the root as some null. Nothing, so no items. The first transaction is what ? So I will label this as I2. I1. I5. What is the count ? How many times have seen this ? Once. what is the next transaction I2,I4 so I start off with null set then I add I2 then I add I4 so this is essentially that.

Okay what is the next transaction I2 anything else of course a lot more I2 I1 so I2 becomes 4, I1 becomes 2, I4 becomes 1. next one is I1 I3 again no not again the first time so what do I do and then what we get I2 I3 so you can see that I can redo the ordering whenever I read the transaction I did not have done it in the first pass so I can do this then where are we t7 and t8 is I2, I1, I3, I5, then the last 29. I2 again then .

So we have our FP tree. So there is only one thing that is needed to complete this. So for ease of navigation, I am going to have pointers from the above table. So where does I2 start here. Likewise where does I1 start here. If there is a second entry of I1 this will connected there what about I3 so I3 is also connected like that then I4 my god it starts looking very scary this is why I wanted the ok.

That is your FP tree. So essentially we constructed this FP three by doing a second scan over the data and if you think about it, so if you will take any path down this tree, okay the prefix of the path, that is the things that come at the beginning of the path will tend to be more frequent than the things that come at the end of the path. This is a whole idea of behind what we have done with this FP tree.

If you take a path from the root to the leaf things that come at the beginning they tend to be more frequent than things that come at the end. So now what we are going to do is we're going to work from bottom up and try to generate an auxiliary data structure for FP tree from which we will generate the frequent patterns.

We can just read off the frequent patterns from this auxiliary data structure. so what is what is it that we do ? We generate something called a conditional tree. We generate something called a conditional pattern base so we typically do this in the reverse order of our table here.

So I will take I5 so I look at all the paths that contain I5 and take the prefixes of that prefix of I5. I will take all the paths where I5 occurs and I take the prefix of that.

Right how many path does I5 occur here ? So I2,I1,I5 . so that is one thing so I will just take the prefix so I2,I1 and it occurs once and then anywhere else I2,I1 , I3,I5. that is I2,I3,I1 sorry I1 sorry I2,I1,I3 there is a conditional pattern base for I5. so I took the things that k now what I will do is I will assume that these are my only transactions and I will create an FP tree.

I will assume that these are the only transactions I have and I will create an FP tree but before I do that I will ignore all the entries in this conditional base which appear only once all the items one item sets that appear only once I will ignore them so I3 appears only once in the whole thing I1 appears twice I2 appear twice but I3 appears only once so I will ignore the occurrence of I3.

Now I will try to create a FP tree with the remaining two transactions so what are the transactions I2,I1 and I2,I1 so my FP tree will look like so I look at the path of a specific frequency so the path here I2,I1 path has a frequency of two and I know that it is followed by I5 because that is how I selected these things so the frequency of I2,I1,I5 will be 2.

We already counted that we countered that by doing multiple passes of the data so now not only will this give me this if to I2,I1,I5 it will give me any frequent pattern in which I5 is a part of regardless of how large the itemset is. so it will give me not only the three item sets frequent itemsets in which I5 is a part of it will also give me the frequent two itemsets in which I5 is part of .

So then I can say that yeah so how do I read the two item sets that I5 is part of just take any prefix in this path and or any combination in this path which has only one thing so not only is I2, I1, I5 is frequent I1,I5 is frequent I2,I5 is also frequent and the frequency is 2. so that is basically done so I have counted all the frequent item sets with effect likewise I can do this for I4.

So what are the frequent what are the what is the conditional pattern base for I4 so this is called the conditional FP tree ok this is the conditional pattern base and the FP tree I construct from the conditional pattern base is called the conditional FP tree and from the conditional FP tree I basically can read off the frequent patterns ok so what is the conditional pattern base for I4 yes we have to look at wherever I4 occurs. that is why you need this data structure go here follow that ok I4 occurred here follow that back so I have I2,I1,I4 occurring once so I2,I1 occurring once.

Again follow that there okay now what can I do I can ignore I1 and construct my FP tree which is even simpler than this conditional FP tree is essentially the frequent patterns are I2, I4 frequent patterns are I2, I4, basically you are done.
all the candidates yeah so the nice thing is now I can ignore I4 and I5 now I want to go to I3 so if you look at it so there is something beyond I3 but I do not have to worry about it if this was part of a frequent pattern I would have already caught it.

When I went back from I5. so when I start now when I go to I3 and I can construct in the conditional pattern base for I3 I have to only look at the prefix above I3 not what comes after I3. so if there was anything it should have been part of that should have been captured already if it hasnt been captured I do not worry about it and that is where we go from I5 to I2 in this case .

So I3 what is the base so start off here so it will be I3,I1,I2 so essentially I2,I1 and the count is two because I3 count was too so the count should be two so far we have only had count of one okay but here the count will be two. in anywhere else I3 occurs here so that is basically I2 anyone else I3 occurs here. how is the conditional FP tree look for this look like for this so again I can read the frequent patterns of this.

So what is it so I2,I1,I3 with the frequency of two I3,I2 with a frequency of 4. I1,I3 with the frequency of two here two here so I1,I3 has a frequency of 4, I2,I3 has a frequency of 4. I2,I1,I3 has a frequency of two. I can just read it off the tree okay so we can check whether that is correct so I1,I2,I3 has a frequency of 4. I1,I2 has a frequency sorry 2, I1,I2 has a frequency of 4. I1,I3 sorry this one is not done I1,I3 has a frequency of 4. I2,I3 has a frequency of 4.

Right so whatever we counted but all the frequent item sets that contain I3 or done in one shot likewise you have to do one for I2 now. I am sorry I1 so what will be the I1 tree. what is the conditional pattern base for I1. we don't have to worry about that way because I know the frequency of one item set I know the frequency of I1 is already six. I know that so basically this is the only thing I will get and so the tree will look like 4. what four times so this path has been taken four times that's what this tells me so that means I2,I1 should have appeared four times so the prefix I2 ending with the suffix I1e would have appeared four times that is basically what the conditional pattern base tells me.

So the conditional pattern base for I1 is just I2:4 for and the conditional FP tree is just one node there is null and then I2:4 and the frequent patterns they yeah so the frequency of I1,I2 is 4 and that is what we get here I1,I2 is 4 we get that and of course the one item set frequency is already given to you by the table

Do I need to do the conditional pattern base for I2, no does not matter because I have taken all the other items so I2 I dont have to actually the process separately so what is the nice thing about this algorithm is that a) I did not have to do a generation of a lot of candidates and then prune things down so I had a way of traversing this tree that just gave me the frequent item sets plus one thing second thing is I just did two passes over the data. all subsequent passes were done on the tree and the tree is somewhat compact assuming patterns actually repeat mean .

If the patterns do not repeat at all and every transaction is a unique subset then you are doomed so you will get a very large tree I mean that'd be slightly compact but still it will still be a large tree but since patterns repeat typically so you are same it is any questions on how this happens so first you do one pass through the data count the frequency of one item sets sort them they do a second pass through the data construct the FP tree put in all this navigation links.

Right and then for each item one item set construct a conditional pattern base and then construct a conditional FP tree and from that you can read off all the frequent item sets that contain that item. make sense any questions on that good point so I have done I have done this I5 so I

basically I computed the conditional pattern base so if you look at a I3 it occurs only once in the entire conditional pattern base .

So I can prune it off so now my conditional pattern base will actually become I2, I1:1 and then I2,I1:1 so I will just create my tree based on that sorry that that is decided by min sup of yeah that is decided by min sup so whichever occurs less than min sup number of times in this I will remove it because it cannot figure in a frequent pattern. so that the one okay you're confused about the one is it no it is the min sup. anything occurs less than min sup because min sup is 2 here.

So it just becomes one so anything that occurs less than min sup number of times so again we can rework the whole thing setting min sup to three and then you would see an interesting things . So I mean all of this will go there will be no frequent pattern of length three so both of this I5, I4 for will go I4, I5 will all go. I4 yeah that is correct , so I5, I4 will all go there will be no patterns that feature I4, I5. in fact they will go from here this table itself.

Because they occur only twice so you only have three entries so we start off with a frequency of three you only have three entries in this table and so you are everything becomes simpler okay that is a good point so if you if you start off with only three entries in this table when you construct the FP tree itself so you leave out I5 and I4 from the transactions there so these entries will not even be made this I5,I4 entries will not even be made at what you will do is if some something get dropped out.

Because the one item set is not frequent I will delete them from the transaction order also so I have sorted them in the decreasing order of this table and whatever is below the threshold I'll just deleted so t1 will become I2, I1 and t2 will become just I2, so that will not even figures in the FP tree construction ok any questions

I bet any applies to Nationals yes there are algorithms that use hashing and in fact take it back.

So the even the algorithms abuse hashing actually give you the count but there are algorithms such that give you an approximation there are so if so many efficient algorithms that give you exact counts nowadays that use hashing lets the I do not know if you should be using approximation but there are other approximation algorithms for this see the more interesting research question to ask now is what happens if I am not just counting elements from not just counting subsets if I am counting data with my additional structure in it wait so that is a more interesting question to ask.

IIT Madras Production

Funded by
Department of the higher education
Ministry of the human resource department
Government of India

www.nptel.ac.in

NPTEL

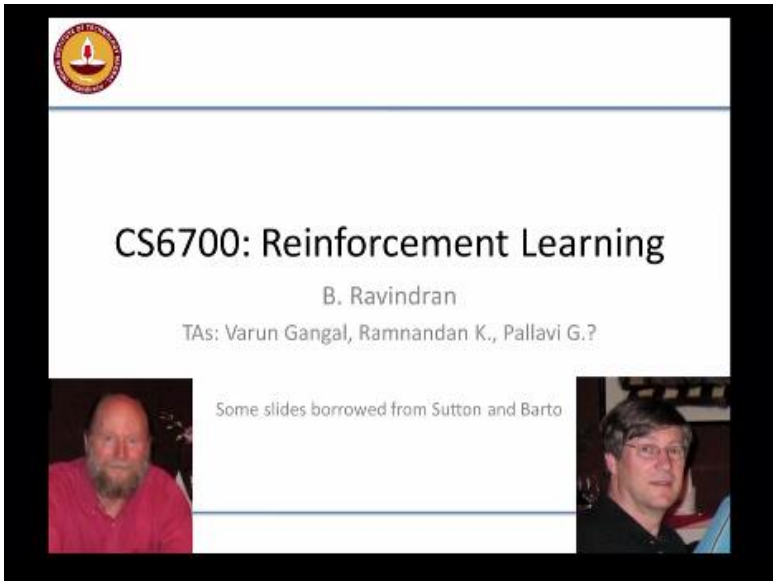
NPTEL ONLINE CERTIFICATION COURSE

Introduction to Machine Learning

Lecture 83
Introduction to Reinforcement Learning

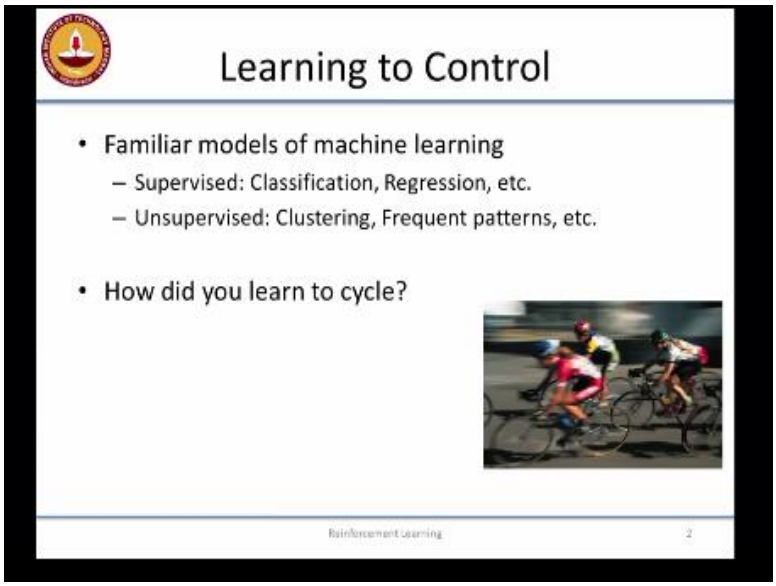
Prof: Balaraman Ravindran
Computer Science and Engineering
Indian Institute of Technology Madras

(Refer Slide Time: 00:15)



Reinforcement Learning is a very different kind of learning than what we looked at in ML. In Machine Learning, the idea was to learn from data, you know you are given lot of data as training instances for you, and then essentially you are trying to learn from those training instance as to what to do. And there were different kinds of problems that we are looking at, so one was supervised learning problem in which you were looking at classification, regression. So, in the machine learning class we looked at learning from data.

(Refer Slide Time: 00:32)



Primarily, so one of the models we looked at was supervised learning , where we learnt about classification and regression, and the goal there was to learn mapping from an input space to a output, which could be a categorical output in the case of classification, it could be a continuous output in this case is called regression . So if you have not been in the ML class do not worry about it.

Because this is just to tell you that RL is not whatever you learnt in the ML class. So if you have not learnt anything in the ML class then you do not have anything to unlearn, so do not worry. So the second kind of learning thing we looked at was unsupervised learning, when there was really no output that was expected of you. Since, therefore there was no supervision, the goal was to find patterns in the input data, I will give you lot of data points, you can find out if there are groupings of similar kinds of data points, can I divide them into segments.

So that kind of thing was called clustering or you are asked to figure out if there were frequently repeating patterns in the data. And so this is called frequent pattern mining or derived problems that was association rule mining and so on so forth.

How did you learn to cycle? Was that somebody tell you how to cycle and then you just followed their instruction. Fell down a couple of times and that automatically made you cycle; you have to actually figure out how to not fall down. So falling down alone is not enough, but you have to try different things. It is not supervised learning; it is really not supervised learning. How much time ever you think? Because now that I have given this talk multiple times, people are getting wise to it. Earlier when I used to ask these people, people say of course it is supervised learning, my uncle was there holding me, or my father was telling me what to do and so on so forth.

And best what did they tell you? Hey, look out, look out do not fall down. So that does not count as supervision . Or keep your body up and some kind of very vague instructions was what they were giving you. Supervised learning would mean that, so you will get on the cycle and somebody will tell you, now push down with your left foot with three pounds of pressure.

And move your center of gravity 3° to the right. So this is something, somebody has to give you exactly what is the control signals that we have to give to your body in order for you to cycle. Then that will be supervised learning , if somebody actually gives you supervision at that scale you probably have never learnt to cycle, if you think about it , because it is such a complex dynamical system and somebody gives you control at that level and gives you input at that level you will never learn to cycle.

And so immediately people flip and say that it was unsupervised learning because here of course nobody told me how to cycle therefore it is unsupervised learning, so if it is truly unsupervised learning what should have happened is you should have watched 100's of videos of people cycling figured out what is the pattern of cycling that they do and get on a cycle and reproduce it.

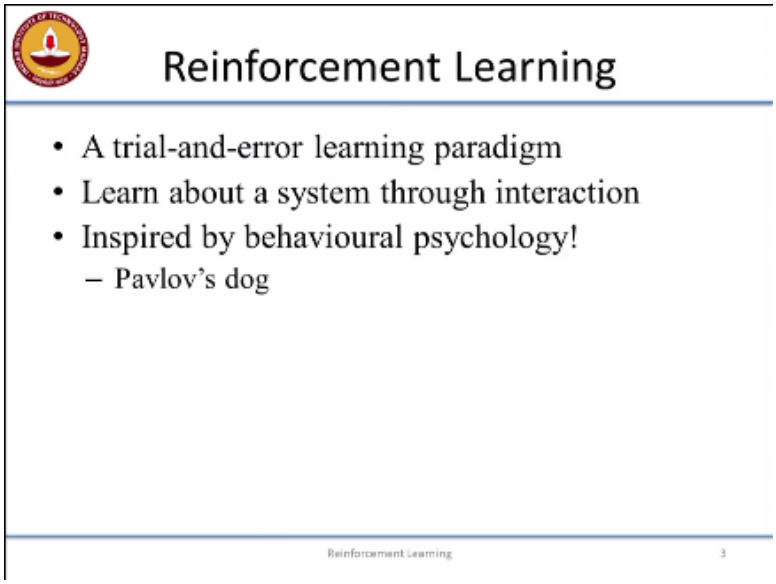
so that is essentially what unsupervised learning would be you just have lot of data and based on the data you figure out what the patterns are and then you try to execute those patterns, that does not work you can watch hours and hours of somebody playing flight simulator, and you cannot go fly a plane, so you have to get on the cycle yourself and you have try things yourself

so that is the crux here so what it is how do you learn to cycle is nether of the above it is neither supervised nor unsupervised it is a different paradigm.

So the reason I always start out my talks not just in the class but in general when I talk about reinforcement learning is because people always talk about reinforcement learning as unsupervised learning. It is just because you do not have a classification error or class label does not make it unsupervised learning. It is completely different form of learning and so reinforcement learning is essentially this mathematical formulation for this trial and error kind of learning.

So how do you learn from this case minimal feedback you know falling down hurts or somebody your mom or somebody stands there and claps when you finally manage to get on the cycle you know that is kind of positive reinforcement when you fall down you get hurt that is kind of a negative feedback how do you just use this kinds of minimal feedback and you learn to cycle so this is essentially the crux of what reinforcement learning is about trial and error. So the goal here is to learn about a system through interacting with the system it is not something that is done completely offline you have some notion of interaction with the system and you learn about the system through that interaction.

(Refer Slide Time: 06:22)

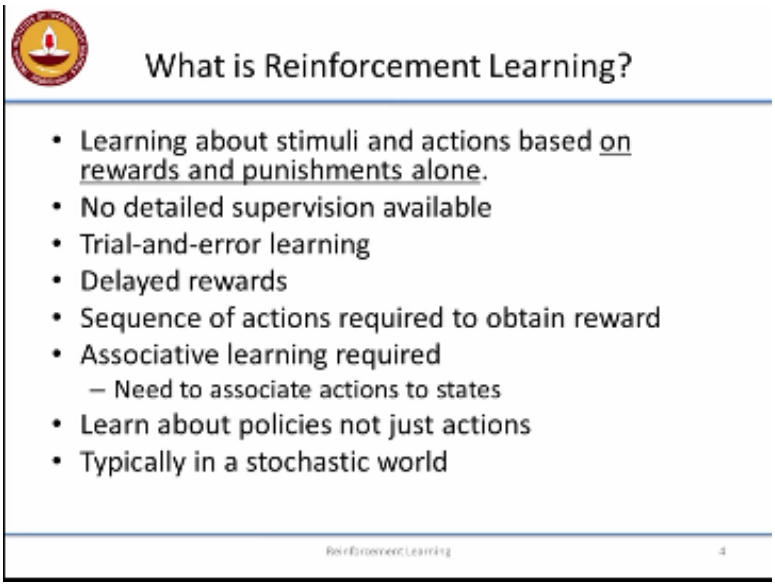


Reinforcement learning originally was inspired by behavioral psychology, so one of the earliest reinforcement systems that are studied with the Pavlov's dog how many of you know of the Pavlov's dog experiment what does the Pavlov's dog experiment. That is called a condition reflects so when the dog looks at the food and starts salivating it is a primary response because there is a reason for it to salivate on the site of food. Any idea why? Exactly. it is preparing to digest the food you know make sure the food is preparing to digest the food it starts salivating.

So, if then now if you think about it hearing the bell and it salivates what is it doing? preparing to digest the bell? No? so when you ring the bell and then serve the food the dog forms an association between the bell and food and later on when you just ring the bell without even serving the food the dog starts salivating in response to digesting the food that it expects to be delivered. So it essentially the food is the pay-off you know the food is like a reward for it and it is learned to form associations between signals in this case which was a bell like an input signal which was the bell and the reward that is going to get , so this is called the behavioral conditioning and so inspired by these kinds of experiments on then more complex behavioral experiments or animals now started to come up with the different theories to explain how learning proceeds .

In fact some of the earlier reinforcement learning papers appeared in the behavioral physiology journals. The earliest paper by Sutton and Barto appeared in brain and behavioral sciences journal. Just to go back I needed to say something about Sutton and Barto. This is a larger audience we can tell that about them. We are going to follow a text book written by Richard Sutton and Andy Barto but more importantly they are also kind of the cofounders of the modern field of the reinforcement learning so in 1983 they wrote a paper, “Adaptive neuron like element that learn the control behavior” or something to their effect . And that essentially kick started this whole modern field of reinforcement learning. So the concept of reinforcement learning like I said goes back to Pavlov and earlier, people have been talking about these kind of behavioral conditioning and learning and stuff but the whole modern computational techniques that people use in the reinforcement learning started by Sutton and Barto.

(Refer Slide Time: 09:41)



So what is reinforcement learning? It is learning about stimuli the inputs that are coming to you and the actions that you can take in response to it learning about the stimuli, only from rewards and punishments , so you are not going to get anything else food is a reward following down and scraping your hand is a punishment, so only from this kinds of rewards and punishments alone there is no detailed supervision available nobody tells you, what is the response that you should give to a specific input.

Suppose you are playing a game, there are multiple ways in which you can learn to play a game. So you can learn to play chess by looking at a board position and then looking at table that tells you for this board position this is the move you have to make, and then you go and make the move. so that is a kind of supervision that you could get you know that gives you a kind of a mapping from the input to the output and essentially you learn to generalize from that. So this is what we mean by detailed supervision. So another way of learning to play chess is just so you have an opponent and you sit in front of him and you just make sequence of moves at end of the move you win you get a reward somebody pays you ten rupees if you lose you have to play the opponent ten rupees, so that's all that happens so that's all the feedback you are going to get . Whether you are going to get the 10 rupees or going to lose the 10 rupees at the end of the game.

So nobody tells you given this position this is the move you should have made that is what we mean by saying learning from rewards and punishments in the absence of detailed supervision. is that clear. And a curial component to this, is trial and error learning because since I do not know what is the thing to do given an input, I need to try multiple things to see what the outcome will be, ?

I need to try different things to see if I am going to get the reward or not? If I do not try different things, I am not going to learn anything at all , so we will I can give you more formal mathematical reasons for why we need all of this as we go on but this is intuitively you can understand this as requiring exploration, so that you know what the outcome is? And there are bunch of things which are also characteristic of reinforcement learning problems one of those is that the outcomes ? the rewards and punishments based on which you are learning can be fairly delayed in time. They did not be temporally close to the thing that caused it. I mean while our playing a game let us say so you might you know now drop a batsman and then he goes on to score like 150 or something like that? So then you lose the match at the end of the day but the event that caused you to lose the match is the dropped catch that probably around the 12^{th} over? Or it could be much more convoluted causal effect? so and how many of you followed cricket, my god it is really losing its popularity yeah put your hands down. I am not going to give you a cricket example then.

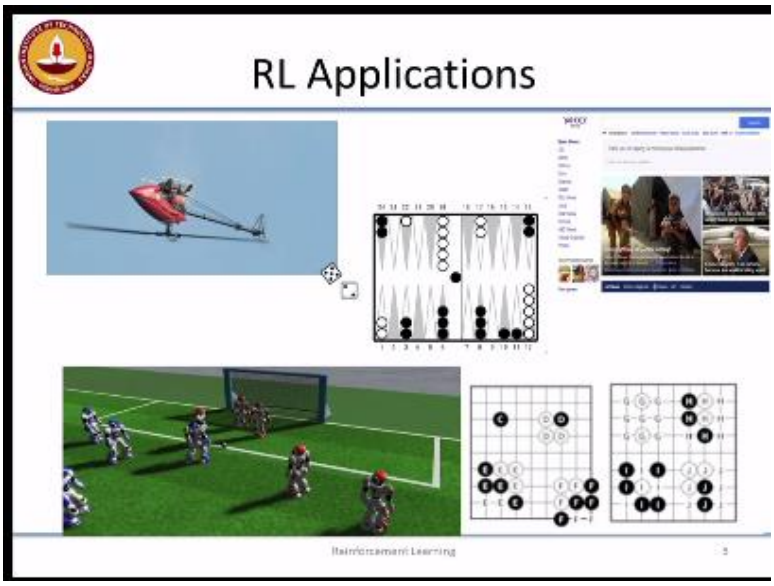
So there a bunch of other things. So we talked about delayed rewards so rewards could come much later in time from the action that caused the reward to happen for example let us go back to our cycling case I might have done something stupid or I might have gone over a stone somewhere well I am cycling at a very high speed and might have been a small stone in the road and that will cause me to lose my balance . And though I will try my level best to get the balance back I might not and I finally fall down and get hurt. That does not mean that what cause the falling down is the last action I tried , I might have desperately tried to jump of the cycle or something like that but that is not what cause the punishment it what cause the punishment happened a few seconds ago when I ran over the stone, .

So there could be this kind of a temporal disconnect between what causes the reward or punishment from the actual reward and punishment so it becomes a little tricky how do you are going to learn those things learn the associations , so quite often you are going to need a sequence of actions to obtain a reward . it is not going to be like a one shot thing? It is going to be a sequence of action to get the reward.

So again going back to the chess example you are not going to get a reward every time you move a piece on the board ? you have to finish playing the game at the end of the game if you actually manage to win you get a reward. so it is a sequence of actions. And therefore you need to learn some kind of a association between the inputs that you are seeing in this case it will be board positions. Or how fast the cycle is moving and how unbalanced you feel and so on so forth. Two actions so inputs that you are getting sometimes which we will call state, and the actions that you take in response to this input that you are seeing , so this is essentially what you are going to be doing when you are solving a reinforcement learning problem, so this kind of associations are essentially known as policies, ?

So what you are essentially learning is a policy to behave in a world ? so learning a policy to play chess or you are learning a policy to cycle ? so this is essentially what you are learning you are not just learning about individual actions ? at all of this happens typically in a noisy stochastic world ? it does makes these things more challenging, so these are all the different characteristics of reinforcement learning problems. So reinforcement learning has been used fairly successfully in a wide variety of applications, ?

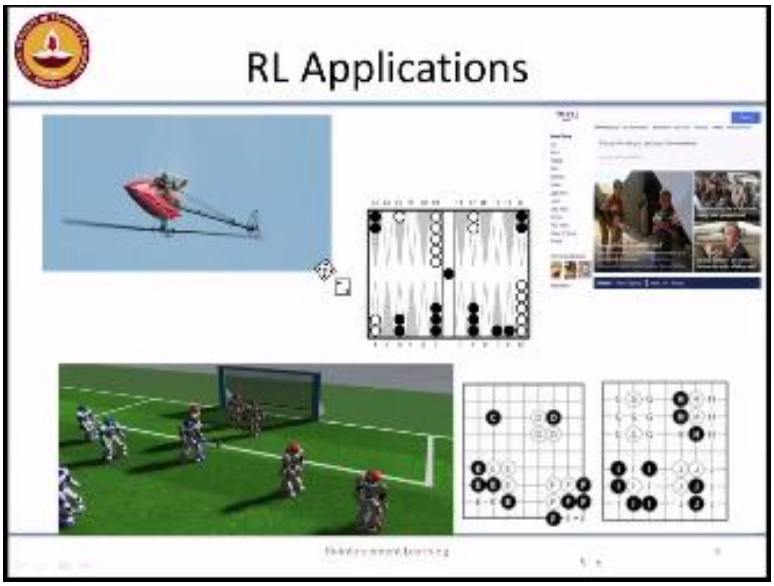
(Refer Slide Time: 15:30)



So you can see a helicopter there ? so that is not a cut and paste error here the helicopter is actually flying upside down ? so the group at Stanford and Berkley which have actually used reinforcement learning to train a helicopter to fly all kinds of things not just upside down and an RL agent can do all kinds of tricks on the helicopter so I will show you a video in a minute.

And it is amazing piece of work, I mean it was considered the show piece application for reinforcement learning. I mean getting such a complex control system to work and it actually could know things at a much finer levels of control than a human being could, well it is after all a machine so you would expect that. but the tricky part was how it learn to control this complex system from without any human intervention. And in the middle, so I have a couple of games there.

(Refer Slide Time: 16:35)



So that is can you see that? So backgammon is like a two player ludo , so you throw the dice you move piece around and you take them of the board , so it is a fairly easy game but then you have all kinds of strategies that you could do with it.

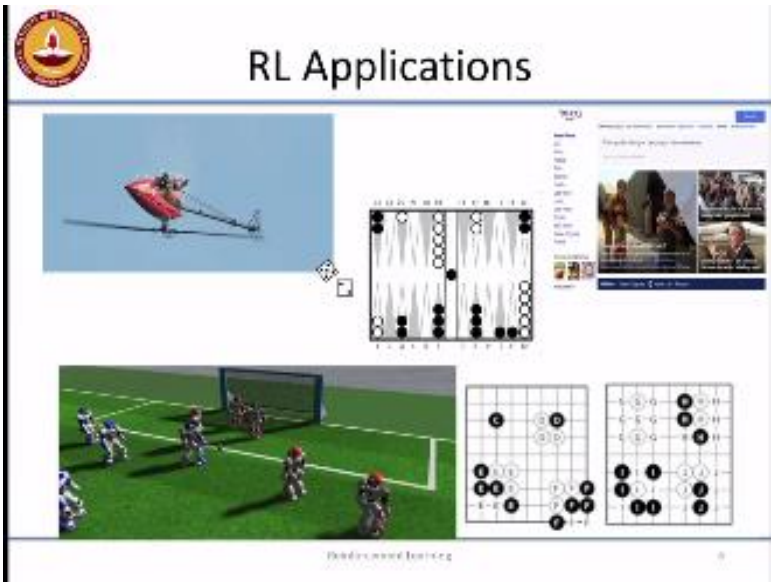
But it is also a hard game for computers to play because of the stochasticity and also because of a large branching factor that is there in the game so at each point there are many, many combinations in which you could move the board pieces around and then there is a die roll that adds in additional complexity, so people are not really getting great results and then there is a person Jerry Tesoro from IBM who came up with something called neurogammon think it was called neurogammon and that was trained using supervised learning under neural network.

And so if you have done it recently, it would have been called the deep learning version of neurogammon or something, because he did it back in the 90's early 90's so is just called neural network version of backgammon and it played really well for a computer program , so that is essentially the best computer program backgammon player at that point, and then Jerry heard about reinforcement learning he decided to train the reinforcement learning agent to play backgammon, .

So what he did was set up this reinforcement learning agent which played against another copy of itself, let them play 100s and 100s of games , rather 1000s and 1000s of games. So essentially what they did was so you trained one copy for like 100 move or 100 games something then you move it here , freeze it and then continue learning with this so essentially what is happening as you learn you are playing against better and better players gradually your opponent was also improving, .

And then this was called self play , so he trained backgammon using self play and it came to a point where the TD gammon as he is called it was even better than the human player of back campaign at that point in the world, so they actually had head to head challenge with the human champion there is a world championship of backgammon you know it is apparently very popular in the middle east and people actually have world championships as a world championship of backgammon and so he challenged the human champion which IBM seems to do a lot . I mean they challenge Kasparov two matches and things like this, so he also challenged yes, Tesaro worked for IBM he should realize , people who spend a lot of resources getting computers to play games well probably be working for IBM you know. So Jerry had this thing and it beat the world champion. So we have reinforcement learning agent that is the best backgammon player in the world, not no more best computer player or anything so we could actually make that claim and there is another game there which snap shot from the game of ‘Go’.

(Refer Slide Time: 20:13)



So people have played go? Oh come on, at least one or two people have played go? People have played Othello? . That's also a very few number. Isn't it one of those free games in Ubuntu? I thought everybody plays that in some point rather the well you rather play Othello than watch paint try you know?

But anyway so, Go is like more a complex version of Othello if you knew it. It is again a very hard game for computers to play because a branching factor is huge and it is actually a miracle that human even play this because the search trees and other things are really complex.

So this is one case which clearly illustrates that humans actually solve problems and fundamentally different way than we try to write down in our algorithm because they seem to be making all kinds of intuitively it is an order to able to be play go. So this person David Silver who currently works for Google Deepmind and but before that he spent some time with Jerry Tesaro at IBM and at some point along the way he came up with this reinforcement learning agent call the TD search that plays go at a decent level. It is still not like master human level performance but it performs plays at a very decent level.

So this is a, what I am pointing out here is things are typically hard for traditional computer algorithm or even traditional machine learning approaches to solve AI has a good success. And here is another example. There are some robots on the bottom left of the screen and so that is a snap shot from the UT Austin robots soccer team called Austin villa and they use reinforcement learning to get there robots to execute really complex strategies.

So this is really cool but the nice thing about the robots soccer application is that they do not use reinforcement learning alone but they actually use a mix of different learning strategies and also planning and so on and so forth which is going on the other studio . So they use a mix of different kinds of AI and machine learning technique in order to get a very, very completed agents it is very hard to beat and they are mean the champions ,I think two or three years running now in the humanoid league. And again hard control problems thinks like how do I take a spot kick you know those are the things for which they use reinforcement learning which it is really hard balancing problem so you have to basically balance the robot on one leg and then swing the other leg so then you take the kick. So, it going to be hard control problem, so they use RL to solve those.

And then up on the top is an application which will probably the one that actually makes money of all these three now all the others that is on essentially on using reinforcement learning to solve online learning. So online learning is a use-case where I do not have the feedback available to me apriori, so the feedbacks coming piece meal. So for example that is the case where we are having new stories that need to be shown to the people that come to my web page and when people come to the page I have some editors will pick like 20 stories for me and from those 20 stories I have to figure out which are the ones that I have put up prominently. And what is the feedback I am going to get?

Nobody tells me what stories that the user is going to like; I mean I cannot have a supervised learning algorithm here. So, from the feedback I am going to get is, if the user clicks on the story I am going to get a reward, if the user does not click on the story I am not going to get a reward, that is essentially the feedback that I am going to get. Nobody tells me anything before hand, so I have to try out things.

I have to show different stories to figure out which one he is going to click on and I have very few attempts to do this in, so how do I do this more effectively? People have done supervised approaches for solving this and it has worked fairly successfully, so but reinforcement learning seems to be a much more natural way of modeling these problems. So not only in these kinds of news story collections, people use reinforcement idea in ads selection. How do I see some of those ads on the sides when you go to Google or some other web page ?

So how those are those ads selected, there might be some basic economic criterion for selecting for slate of ads. So here are the 10 ads which will probably give me the pay off and then you can figure out, which 3 of those ten am I going to put it out over here and things like that, you could use the reinforcement learning solution for selecting those. Ofcourse, this whole field called computation advertising, is a lot more complex than what I explained; But RL is a component in computational advertisement as well.

(Refer Slide Time: 26:21)

IIT Madras Production

Funded by

Department of Higher Education

Ministry of Human Resource Development

Government of India

www.nptel.ac.in

NPTEL

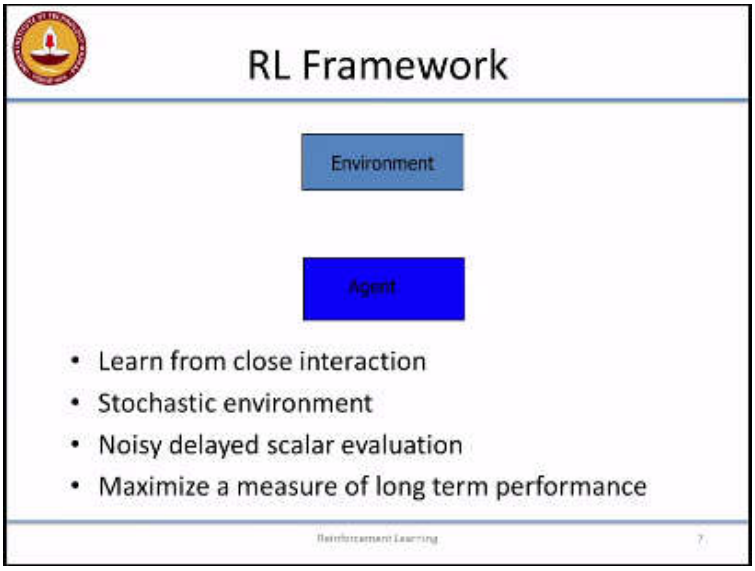
NPTEL ONLINE CERTIFICATION COURSE

Introduction to Machine Learning

Lecture-84
RL Framework and TD Learning

Prof: Balaraman Ravindran
Computer Science and Engineering
Indian Institute of Technology Madras

(Refer Slide Time: 00:15)



The crux in the reinforcement learning is that the agent is going to that is the learning agent it is going to learn in close interaction with an environment so the environment could be the helicopter it could be the cycle and or it could be your backgammon board and your opponent all of this could constitute environment a variety of different choices so you sense the state in which the environment is in you sense the state of the environment and you can figure out what is the action that you should take in response to the state .

So in apply the action back to the environment this causes a change in the state so now comes the tricky part so you should not just choose actions that are beneficial in the current state but it should choose actions in such a way that they will put you in a state which is beneficial for you in the future just capturing the queen of your opponent is not enough in the chess that might give you a higher reward but it might put you in a really bad position so you do not want that you really want to be looking at the entire sequence of decisions that you are going to have to make.

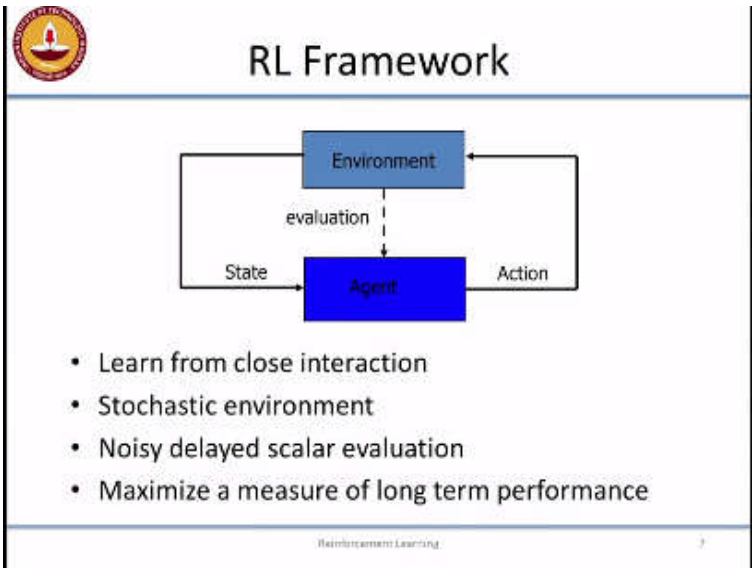
And then try to behave optimally with respect to that . So what we mean by behave optimally in this case we are going to assume that the environment is giving you some kind of an evaluation it is like falling down hurts when or capturing a piece maybe gives you a small plus point five or winning the game gives you like hundred so every time you win every time you make a move or every time you execute an action you did not get a reward or you did not get an evaluation from the environment .

So it could be just zero it could be it could be nothing so I should point out that this whole idea of having an evaluation come from the environment is just a mathematical convenience that we have here but in reality if you think about biological systems that are learning using reinforcement learning all they are getting is the usual sensory inputs so there is some fraction in the brain okay that sits there and interprets some of those sensory input as rewards or punishments .

So you fall down you get hurt I mean that is still a sensory input that is coming from your skin or somebody pats you on your back that is still a sensory input that comes from the skin and it is just another kind of an input so it could choose to interpret this as a reward or this as a collision with an obstacle something is brushing against my shoulder let me move it or you can just take it as somebody is patting my back so I did something good .

So it is a matter of interpretation so this is a this whole thing about having a state signal and having a separate evaluation coming from the environment is a friction there is created to have a clear cleaner mathematical model but in reality things are a lot messier you do not have such a clean separation and like I said so you have a stochastic environment.

(Refer Slide Time: 03:18)



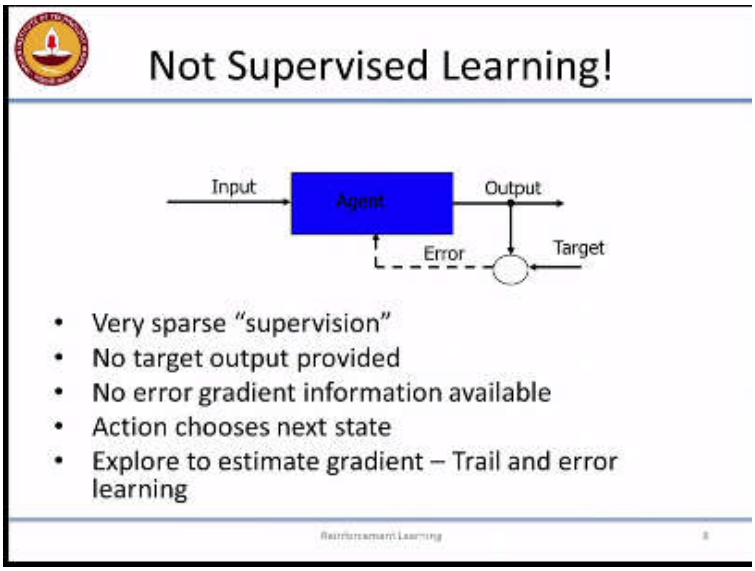
You have delayed evaluation noisy so the new term that we have added here is scalar the new term we have added is scalar so that is one of the things with the classical reinforcement learning approaches he said I am going to assume that my reward is a scalar signal so have we talked about getting hurt and having food and so on so forth what do all of this will happen mathematically is I will convert that into some kind of a number on a scale .

So getting hurt might be minus 100 getting food might be plus 5 winning the game might be plus 20 capturing a piece might be +0.5 or something like it so I am going to convert them to a scale and the goal is now know that I have a single number that represents the evaluation the goal is now to get as much as possible of that quantity over the long run okay, make sense . So if you have questions doubts stop me and ask.

So mathematically a scalar is easier to optimize not necessarily I am just talking about so it as like a cost function if you want to think about it in terms of in terms of control systems so this is like a cost and I am trying to optimize the cost all and so for the cost is going to be vector value and then I have to start trading off one direction of the vector against the other so which

component of the vector is more important so then it get into all kinds of super at optimality and of questions so it is not really clear what exactly is optimal in such cases so here again let me emphasize it is not supervised learning .

(Refer Slide Time: 05:11)



In supervised learning this is essentially what you are going to see there will be an input and there will be an output that you are producing and somebody will be giving you a target output okay so this is what you are supposed to produce and essentially compare the output you are producing to the target output and we can form some error signal and you can use that error in order to train your agent .

You can try to minimize the error you can do gradient descent on their work and do variety of things you can try to train the agent so here I do not have a target I do have to learn a mapping from the input to the output but I do not have a target and hence I cannot form an error and therefore my trial and error becomes very essential see if I have errors rate I can form gradients of the errors and I can go in the opposite direction of the gradient of the error and then that gives me some direction in which to change my parameters and that constitute the agent all major is going to be described in some way the error gives me a direction but now since I do

not know a direction so I just I do something I get one evaluations I do not know that the evaluation is good or bad so think of writing an exam I do not tell you the answer I just tell you three and so what do you do now do happy with answer should we change it should it change it in one direction or should he change it to the other direction.

See what makes it even more tricky is I do not you do not even know how much the exam is out of so when I say 3 it could be three out of three it could be three out of 100 , so it could be any of these things so you don't even know whether, three is a good number or a bad number so you have to explore to figure out A if they can get higher than three or three is the best the second thing is if I can get higher than three how should I change my parameters to get to become higher than three.

Let us I have to change my parameters a little bit that way okay I have to change the parameter rise a little bit this way so if I am cycling wait I have to push down a little harder on the pedal okay I will have to push down a little softer on the pedal to figure out whether I am staying balanced for a longer period of time or not I do not know that otherwise unless I try these things I would not know this is why the trial and error part.

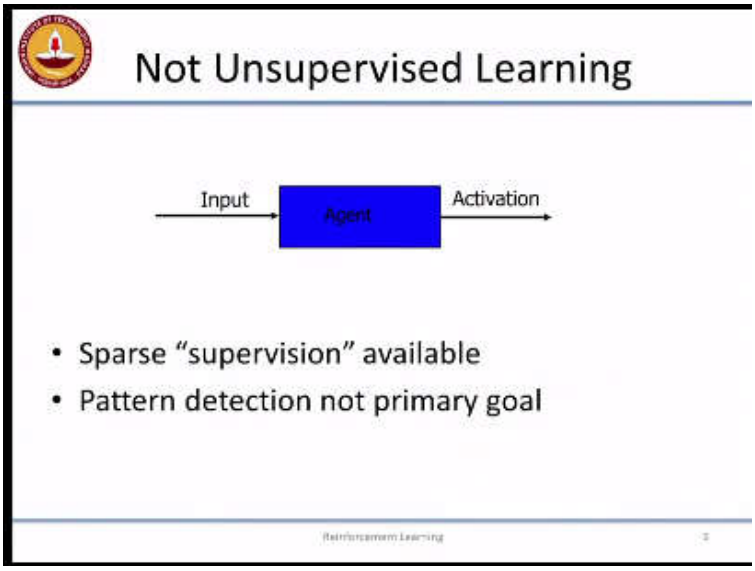
So if I pushed on a little harder and I stay balanced maybe I should try pushing down even more harder next time so maybe that will make it better and then there might be some point where it too poor so I need to come back so this is how things which you have to try unless you try that you do not even know which direction you have to move in so this is much more than just the psychological aspects of trial and error there is also a mathematical reason if you want to adopt my parameters I need to know the gradient okay so that you need to yeah.

The reward is the one that you know that gives you the evaluation for the output so herein the supervised case the error is the evaluation for the output of the error is 0 then your output is perfect but then the way of gauging what the error is because you have a target which you can compare and from there you get the error so in the reinforcement learning case the evaluation is directly given to you as the evaluation of the output it is not necessarily comparing against a target value or anything you do not know how the evaluation was generated that you just get an

evaluation directly so you just get some number corresponding to the output and so maybe I should have done put an arrow from the top saying evaluation comes in from there.

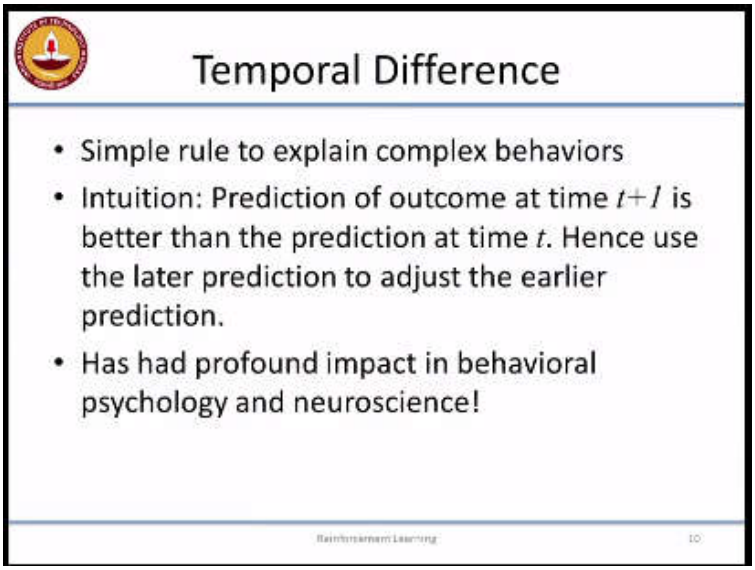
But that is exactly where it is coming let us substitute for the error signal but it is just that you do not know what the evaluation is of course the way differs from the error is minor differences you typically tend to minimize error but you tend to maximize evaluation it is also not unsupervised learning so unsupervised learning has some kind of an input .

(Refer Slide Time: 09:07)



That goes to the agent and then it figures out what are the patterns for thee in the input here you have some kind of an evaluation and you are expected to produce an action in response to the input it is not simply pattern detection so you might want to detect patterns in the input so that you know what is the response to give but that is not the primary goal but in unsupervised learning the pattern deduction itself is the primary co so that is the difference .

(Refer Slide Time: 09:41)



So here is one slide which I think is kind of the soul of reinforcement learning it is called temporal difference so I will explain a little more detail and in a couple of slides but the intuition here so if you remember the Pavlov's dog experiment what was the dog doing it was predicting the outcome of the bell you know if the bell rings there is an outcome that is going to happen it is predicting the outcome which is food is going to happen and then it was reacting appropriate to the outcome .

So most of reinforcement learning you are going to be predicting some kind of outcome that is going to happen since I am I going to get a reward if I do this or if I am I going to not get a reward I am I going to win this game if I make this move what am I not going to win this game all so I am always trying to predict the outcome. The outcome here is the amount of reward or punishment I am going to get this is essentially what I am trying to predict at every point .

So the intuition behind the what is called temporal difference learning is the following so the prediction that I make at time t+1 okay of what will be the eventual outcome let us say I am playing a game I am going to say I am going to win now I am very sure I am going to win down, . So I can say that with a greater confidence closer to the end of the game then I can at the beginning of the game so I have all the pieces set up and if I am going to sit there then and say I

am going to win the game it is most probably visual thinking but then you have played the game for like 30 minutes or something and there are like five pieces left on the board.

Now I am going to say I am going to win the game now I say I am going to win the game that is a much more confident prediction then what I did at the beginning so taking this to the extreme so the prediction I make at t + 1 is probably more accurate than the prediction I make at t , the prediction I make it t + 1 is more accurate than the prediction I make it t, so if I want to improve the prediction I make at t, what can I do?

I can look go forward in time then basically go to the next day let the clock tick over and see what is the prediction I will make a time t+1 with additional knowledge I am getting , I would have moved one step closer to the end of the game so I know I know its little bit better about the game I do not know how the game is proceeding I know I can may now make a prediction about whether i will not lose .

And use this go back and modify the prediction I make it time at time t, and t I think there is a possibility of say probability of 0.6 of me winning the game okay and then we make a move then I find out that I am going to lose the game with a very high probability then what will I do is I will go back and reduce the probability of winning that I made at time t so instead of 0.6 I will say okay maybe 0.55 or something .

So next time I come to the same state as I was at time t, I would not make the prediction of 0.6 I will say 0.55 that is essentially the idea behind component difference learning so it has a whole lot of advantages we will talk about it a couple of slides down but one thing is that the significant impact in behavioral psychology and in neuroscience so it is widely accepted that animals actually use some form of temporal difference learning.

And in fact there are specific models that that have been proposed for temporal difference learning which seemed to explain some of the neuron transmitter behaviors in the brain yeah, no see at this point I will be making a prediction about what is the probability of winning and it

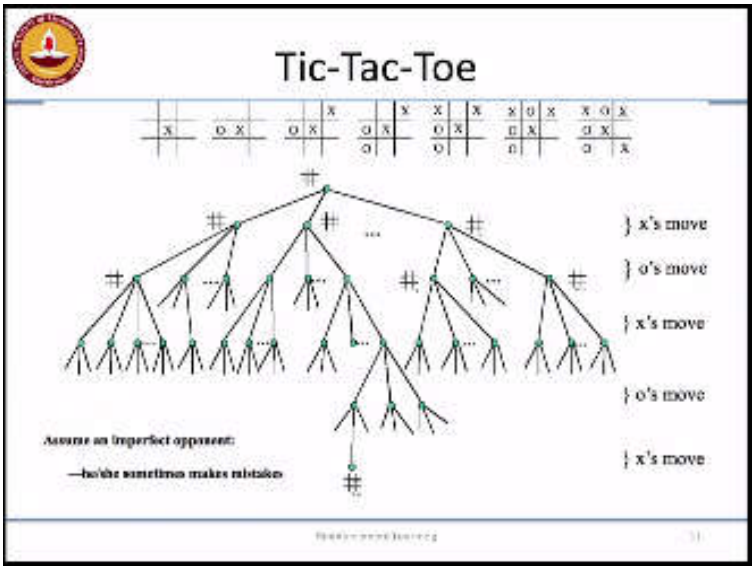
could be for each of the moves if I make this move what is the probability opening if I make this move what is the probability of winning.

Let us say I make move two okay and then I go I see a new position the opponent response to it and then I decide oh my god this is a much worse move than I thought earlier, so what I do is I will change the prediction I make for move two in the previous state you see that the other moves will not be affected because the only move I took was 2 only about that move I have additional information therefore I can go back and change the prediction I make for move 2 alone.

So you can still have the 10 moves so they are not changing any of that yeah, seeing the prediction is not like I mean if an ideal world you should be able to take back a bad move , except if it is a parent playing with the kid I do not think those things are allowed , in fact I when I play with my son we have sometimes had to rewind back all the way to the beginning it will probably be me asking to do the rewinding not him, because he will be dropping me in someone of those games but yeah, otherwise you cannot you just make the change the prediction.

So next time you play the game it will be better at it, it not for that game well basically you have messed up or you did well I mean so whatever it is yeah, okay I hope I was not too boring in somebody fell over , that is known to happen yeah, so people sleep and I actually had a person sleep and they fall off the chair once.

(Refer Slide Time: 15:24)



Yeah, I still cannot get over this okay, there is one time was going to teach a class , and I said was entering the class one person was leaving the class I said hey, what are you doing I suppose me in my class he said no, no, I feel very sleepy I cannot and I do not care if you are going to sleep this get back to the class , and he looked at me for a minute shortly I said okay, and they walked into the class went to the last place actually lie down on the bench on and I going to sleep okay, and he recently sent me a friend request okay, coming back to looking at the RL .

So listen looked at tic-tac-toe , how many of you have played tic-tac-toe good, even you put your hand up okay, good so in tic-tac-toe so you have these board positions , and so you make different moves, so in the first this is what I have drawn here is called a game tree , so I start off with the initial board which is empty , and there are how many possible branches there for people making moves nine possible branches , for excess move there are nine possible moves I have nine possible branches and then for each of these I will have like eight possible I am not sure this is the TV and for each of this i have eight possible branches and they keep going, .

So what we are going to be doing is essentially trying to this formulate this as a reinforcement learning problem so how will you do this as a reinforcement problem , so I have all these board positions , let us say x is the reinforcement learning agent and o is the opponent , so initially given a blank board I will have to choose one among nine actions , so there state that I am going

to see is this the X's and O's on the board , and the moves I will be making are the actions . So in the initial position I have nine actions I make that do I get any reward not really there is no natural reward signal that you can give.

Essentially the reward that I am going to get in this case is if at the end of the moves if I win I will get a 1 if I do not win I get a 0, and if I win I get a 1 if I do not win I get a 0 , so what is going to happen is I am going to keep playing these games multiple times , and at each point yeah, okay, so there is a note here so what is it note say. You have to assume it is an imperfect opponent , otherwise there is no point in trying to learn tic-tac-toe why.

We will always draw and the way we have set up the game you are indifferent between drawing and losing so you learn nothing I mean basically, so you will not know even learn to draw okay, you will just learn to nothing, basically we learn nothing because you can never win , so you are never going to get a reward of 1 so you will just be playing randomly, so it is this is a bad, bad idea so let us assume that you have an agent that is imperfect , that makes mistakes so that you can actually learn to figure out where the agent makes mistakes, where the opponent makes mistakes and learn to exploit those things, okay, .

So your states are going to be this board positions as you can see we give you see a game that has been played out on the top of the slide , and the actions you take or in response to those board positions and finally at the end of the game and if you win you get a 1 if you do not win you get a 0 , Sir, in case like I mean does it have to be a binary sort of a reward system I mean could you have a scale whether there are three parameters you lose a 0 if you draw 1 by privilege 1.

Sure, you could even know other things like if you win it is 1 if you lose its -1. Yeah, you possibly could but you probably have to pay a lot of games because the perfect opponent it is almost impossible for you to start getting any feedback in the beginning , you will always be losing so it is going to be hard for it learn but you will eventually learn something yeah, it will take a lot of mozes level to learned something go back, so if I say that at every point, so we are

learning like at a particular stage the probability of winning and like it is what I am going to US state so you are storing information for each and every state that you have entered .

So how will it be different from exploring the proper next state space every time because after you have done let us say a thousand games or million games you would rather explored a lot of states I will have to store for each state the probability of you winning at that point. Yeah. And all that so I will that be different from exploring it again.

I know the probability of winning program why would I have to close, this still I am not even totally how you are going to solve it okay, let me explain that and then you can come back and ask me these questions, okay if you still have that okay, quit. So what the way we are going to try and solve this game is as follows , for every board position I am going to try and estimate the reward I will get if I start from there and play the game to the end , every board position I am going to look at the word I will get if I start from there and play till the end.

Now if you think about it what will this reward connected , so if I win from there I will get a 1 if I lose from that or if I do not win from that I will get a 0 , when I say what is the reward I expect to get starting from this board position , it is essentially this average over multiple games, it some games I will win some games I will lose, or I will not win like some games I win some games I will not win so what will this expected reward represent after having played many, many, many games.

The probability of winning , the reward is going to represent the probability of winning in this particular case , if the reward had not been 1 , if it had been something else if it had been +5 that you would have been some function of the probability of winning , half it has been +1 for winning -1 for loosing and 0 for draw well it is something more complex is no longer the probability of winning , it is the gain I expect to get , how what fraction of games I expect to win over the fraction of games I expect to lose or something like that, so it becomes a little bit more complex.

So there could be some interpretation for the value function but in general it is just the expected reward that I am going to get starting from a particular board position okay, so that is what I am trying to estimate , that is assume that I have such a expectation well defined for me , as you say I have such an expectation well defined, . Now I come to a specific position let us say I come to this position here , let us say I come to this position.

How will I decide what is the next move I have to make sorry, whichever next state has the highest probability of winning so I just look ahead to see okay where if I put if the x here , if I put the X here what is the probability of winning, if I put the X here what is the probability of winning, if you put takes here what is the probability of winning , I do this for each one of these , and then I figure out whichever has the highest probability of ending and I will put the x there, .

So that is how I am going to use this function does it make sense, yes, it is very important so this is this is something which issued understand this is the crux of all reinforcement learning algorithms , I am going to learn this function that tells me if you are in this state , if we play things out to the end what will be the expected payoff that you will get , whether the rewards or punishment or cost whatever you want to call it what is the expected value you are going to get and I want to behave according to a this learnt function.

So when I come to a state I look ahead figure out which of the next states has the highest expectation and then go to the state okay, great how do I learn this expectation. What is the simplest way to learn the expectations, this is especially keep track of what happens, essentially keep track of the trajectory through the game tree , you play a game you go all the way to the end .

So you keep track of the trajectory and if you win , you go back along the trajectory and update every state that you saw on the trajectory you update the probability of winning , it just increase it a little bit or you come to the end of the game and you found that you have not 1 , you go back along the trajectory decrease the probability of winning a little bit, .

Alternatively you can keep the history of all the games you are played so far , after every game has been completed you can go back and compute the average probability of winning across the entire history of all the games in which you saw that particular position , make sense thus easiest way of estimating this probability, .

But the problem with this is a you have to wait till the game ends , or you have to store the history of all the games you have played try to means all of these could be potential drawbacks okay, you can get around the history part by coming up with an incremental rule but the main difficulty here is you will have to wait all the way to the end of the game , before you can change anything along the way so tic-tac-toe was easy is like how many moves can you make in tic-tac-toe at best 4 , the fifth one is determined for you, .

So it is basically four choices that you can make , so and that is easy enough to remember , you can always wait till the end of the game and then you can always make the updates , what if it is a much more complex situation , what if you are playing chess, maybe you can wait till the end, so what if you are cycling maybe you can wait till the end exactly we do not know , this it depends on where you are cycling if you are cycling learning to cycling 90 meters it is fine, when you are learning to cycle somewhere on the Sardar Patel road you do not want even think about what end is there , so this is there are some tasks for which you really like to learn along the way, .

So this is where TD learning comes into comes to help , I do not think I have it slide anyway and I am not using the fancy thing where I can draw on the projection, so let us see if I can do it here . Suppose I have come here , and from here I have played at this point I know the probability of winning is say point 4 , so I came here by making a move from this position, so I said here late and we made a we know that the probability of winning from here is a point 3 .

But I made the move from here to come here , but here I had thought my probability of winning was let us say point 6 then I thought my probability of winning was of point 6 , but then I looked at my next states and I found that the best one was point 3 somehow , so I went there . But now since the best I can do from here this point 3 me saying point 6 here there is something wrong ,

so I should probably decrease the probability of winning from here . So why could it be, why could it have happened that I thought that was point 6, but the best among the next was point 3, the thing is.

So that whenever I came to came through this part maybe I won before it so happened that when I went through like this initially I would have gone through like this and played the game and i am the examples I drew I might have actually won some of those games . So I would have change this 26 but it is possible for me to get here by playing a different sequence of moves also .

So for example to come here I could have put the X first here and then here or I could have put the X like I did here I put the X first here and then here, that either way I would have reached this position , so there are many combinations in which I could have reached the same positions it just to be nice to these guys . To reach here there are different orders in which I could have put the O and the X here we have showing a cell specific order the O first 1st put here then put here that the x was first put here. Then put in it could very well be I put the X first here in the war first here and then I put the X here in the O here .

The multiple ways in this thing goes reach so sometimes when they play those games I lost, all sometimes and it played these games I won therefore it turns out that for due to some random fluctuations , so sometimes i win when I go through this place specific point and that is what I have a higher evaluation of winning but when I went through the other paths I had a lower evaluation of winning . But we know that really does not matter what path you went through in tic-tac-toe . Once you reach that point what is going to happen further is determined only by that point.

So what i can do now is take this 0.3 you should update that 0.6 down, so I am very confident here I think I will win with the probability of 0.6 but the best probability I have from the next stage is 0.3, therefore here I should not be so confident good point, that depends on the how stochastic your game is . So if you are game has a lot of variability then you do not want to make a complete you know commitment to a 0.3 so you might want to say ok now let me move it little

bit towards 0.3 . But if it is a more or less a deterministic game then you can say okay 0.3 yes sure let me go to all the way to, in the difference on the yeah it is misleading it is called game tree actually.

But it is a game graph in this case yeah, so as I said kid when this is this is an instance of temporal difference learning, so how while I use the thing to update this is called temporal differential learning okay. So there is one other thing which I should mention here if I always take the move that which I think is the best move now, let us talk about it I start tab I have never played tic-tac-toe before , so I play the game I play it once I get I get to the end I win. So now what I do I go back whether I am using temporal difference learning or waiting till the end up dating whatever it is I change the value of all the moves I have made in this particular game .

So the next time I come to a board position what am I going to do? I look at all possible outcomes everything except the one that I have played will have a 0 and the one that I have played will have something slightly higher than 0, I am going to take that, then in fact it will be like how many of you watch the movie Groundhog Day? It will be like Groundhog Day I will be playing I will be playing the same game again and again because that is what happened to give me a win in the first time around . That but that might not be the best way to play this .

So I need to explore , so I need to explore multiple options so I should not be always playing the best move I should always be paying the best move I need to do some amount of exploration, so that I can figure out if there are better moves than what I think is currently the best move . So tic-tac-toe there is inherently some kind of noise if your opponent is random but if an opponent is not random and if operand is also playing a fixed rule and if you are playing greedy, then you will be just plain a very small fraction of the game tree and you would not have explored the rest of the outcomes .

So you have to do some amount of things at random so that you learn more about the game . So here is a question for you, when I am estimating this probabilities of winning , let us say I have reached here I look down and the action that gives me the highest probability of winning say gives me a probability of say 0.8 what I want to explore so I take an action that gives me a

probability say 0.4 okay. So I will go from here to another action that has a probability 0.4 that another board positions that has a probability of 0.4 of winning. So should I use this 0.4 to update this probability or not.

No why? that you are questioning the whole TD idea and you are exploring you should probably wait for the or not just ignore it okay, any of any other answer because you are good or a bad move will be found out I have to update the value of that move I agree. Do I update the value of winning from the previous board position was the question so that 0.4 I will have to change but do i change the 0.8, that was the question the 0.8 was a probability of winning from here I look or whatever. So probably say I had a probability of winning of 0.6 from here I look at the bottom and the best probability of winning says 0.8.

But then I take because I am exploring I take an action that has a probability of winning of 0.4 all the question is do I go back and change the 0.6 towards 0.4 or do I leave the 0.6 as it is? Sorry that one where I am exploring , I mean this is we will be necessarily be less than 0. 8 this will be 0. 4 will be 0.6. So the question here is 1 way of arguing about this is to say that, a if I am playing to win I will play the best action from here and then the best action says 0. 8 therefore I should not penalize it for the bad action which is 0. 4.

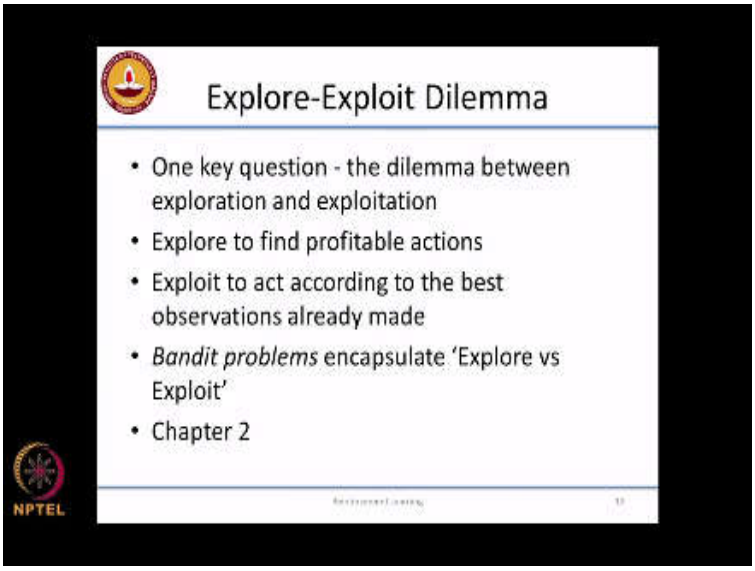
Which I did to learn more about the system and that is one way of thinking about it another way of arguing is to say that hey, no this is how I am actually behaving now . So I should give you the probability of winning about I about the current behavior policy , this should not be some other ideal policy should be about to what I am behaving currently and therefore I should update it . So which one is correct first or the second questions? But this is something these are this is like I said ask you to think about the whole tic-tac-toe thing and many of these answers have relevance later on.

In fact there are 2 different algorithms one does option one does option two , so there is no answer or wrong answer answer is depends yeah, so yeah so this is a different things that you can think about in this but I told you about 2 different ways of learning with tic-tac-toe one wait till the end and figure out what the probabilities will be, the other one is keep adapting this as

you go along and both cases you not explore that is it to keep out here in both cases you have to explore otherwise will not learn about the entire game.

So this is where the Explore exploit thingy comes in okay yeah. Great question different algorithms deal with indifferent way that is one of the crucial questions that you have to answer in RL. So it is called the explore exploit dilemma , so you have to explore to find out which is the best action and you have to exploit.

(Refer Slide Time: 37:19)



Whatever knowledge you have gathered and you have to act according to the best observations already made , so this is called exploitation . So the key one of the key questions is when do you know you have explored enough should I explore now or should I exploit now, this is called the explore exploit dilemma and a slightly simpler version of reinforcement learning called the Bandit problems okay. Some carefully called bandit problems they of course he is an expert on bandit problems here you can the Bandit problems encapsulate.

This explore exploit dilemma lot of people are turning and looking at a noticeable but, so this will ignore a whole bunch of other things like the delayed rewards you know the sequential

decisions and other things. Even in the absence of all of these other complications that even if I say that you are all your problem is you have to take an action and you will get a reward okay your goal is to pick the action that gives the highest reward. I will give you 10 actions you have to pick the action that gives you the highest reward , but the problem is you do not know what is the probability distribution from which these rewards are coming .

So you will have to do some exploration I have to actually do every action, at least once okay to know what will be the reward even if they are deterministic . So I cannot say which is the best action before I try every action at least once, if it is deterministic it is fine I can just try every action once and I know what is the payoff . But if it is too stochastic I have to try every action multiple times how many times you have to try it depends on the variability of the distribution.

IIT Madras Production

Funded by
Department of Higher Education
Ministry of Human Resource Development
Government of India

www.nptel.ac.in

NPTEL

NPTEL ONLINE CERTIFICATION COURSE

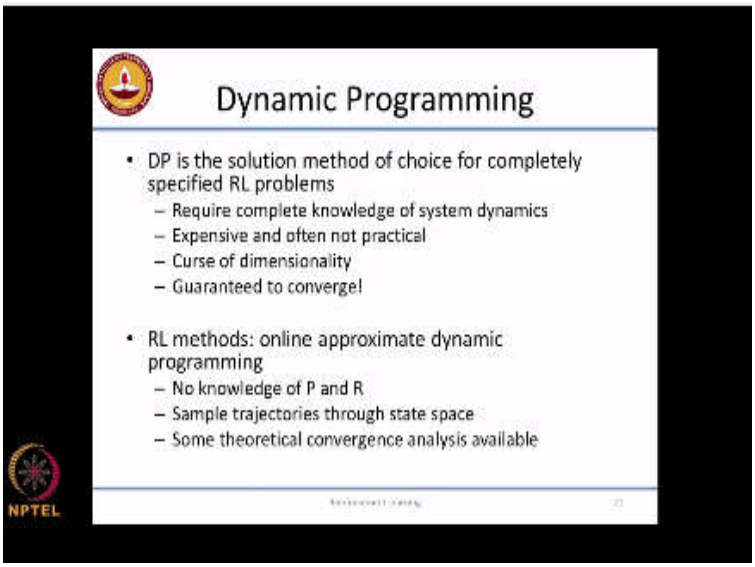
Introduction to Machine Learning

Lecture-85
Solution Methods & Applications

Prof. Balaraman Ravindran
Computer Science and Engineering
Indian Institute of Technology Madras

There are two classes of algorithms that we will be talking about. So one of them is based on what is called dynamic programming.

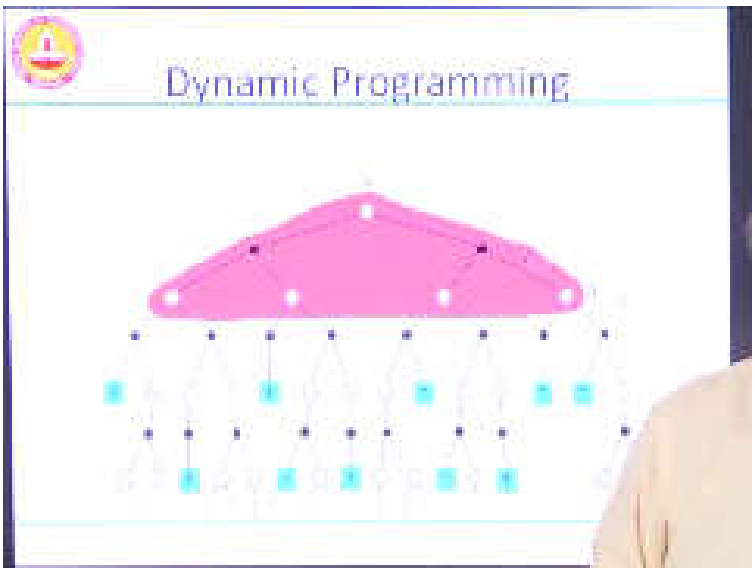
(Refer Slide Time: 00:29)



Now the essential idea behind dynamic programming is you will be using some kind of repeated structure in the problem . So I have to solve the problem more efficiently , suppose I have a solution that I can give for a problem of say size n . Then I will try to see if I can use that for defining the solution for the problem of size n+ 1, so some kind of a repeated sub structure in the problems . The very rough way of describing what dynamic programming is . So for example

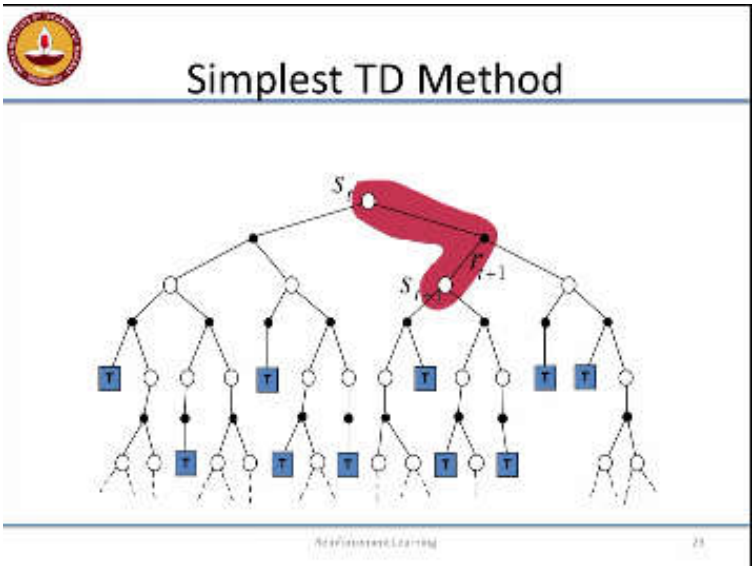
one way of thinking about dynamic programming is I have this game tree , so I look at the values of winning or the expectations of winning from all of these steps . I will use these in order to compute the value of winning or the probability of winning from the state .

(Refer Slide Time: 01:49)



So if you think about it if from here if I am going to take say n steps, from here how many steps will be expect me to take? N minus 1 steps , so I look at the probability of winning when I when I have only n - 1 steps left , I will estimate that first I will use that solution for estimating the probability of winning when I have n steps only n -1 steps left I will estimate that first, I will use that solution for estimating the probability of winning when I have n steps left . So that is essentially the idea behind dynamic programming and so all you do now is instead of having the entire outcome.

(Refer Slide Time: 02:20)



And using that for estimating the probability of winning here I am going to just use one step that I take through the tree, I use this what happens in this one step I will use that in order to update the probability of winning here. all s instead of using the entire outcome as a dynamic programming in reinforcement learning methods we will be using samples that you are getting through the state space okay this is the TD method.

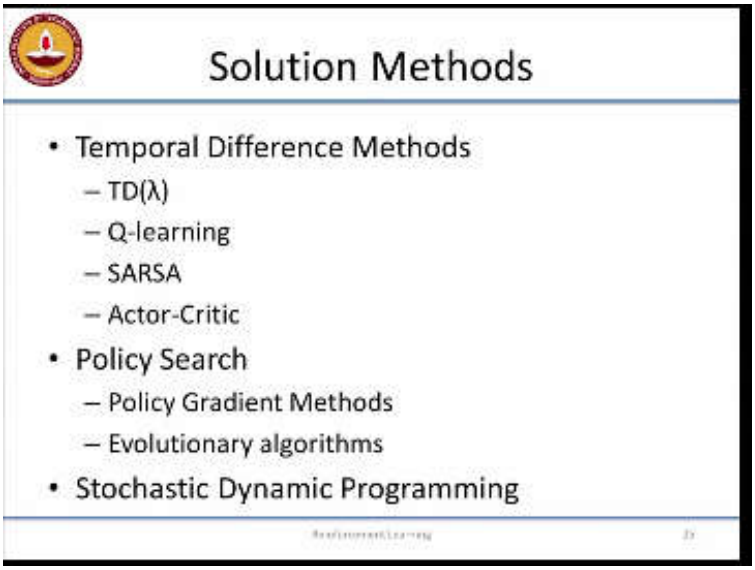
In the other method I explained to in for tic-tac-toe what would you do your sample will run all the way to the end and you use that to update. So this is the two different ways of using samples here, so f T will be determined by the value of as p +1 so the value of s t + 1 should be computed for all and so depends on the steepest to again that should be computed first so would not this be the same thing as exploring the whole all the way down and then computing this if you are doing it for the first time the first time down the tree there will be no updates actually because st +1 is also be 0 st will also be 0 the first time we go down.

There will be no updates but once you reach an end then from there you start updating the previous day, so the next time you go down the tree then it will keep going further up whether in what the game have lost the game I actually I am taking the exact outcome that happened in that particular trial , at that particular game and I used that to change my probabilities but here I am not just taking the outcome of that particular game I am looking at the expected value of

winning from the next board position . So if I wait there all the way here and I say I won and it take this in update st then I will be only updating it with the fact whether I won or not okay but if I am updating it from st + 1 see I could have reached st multiple ways before .

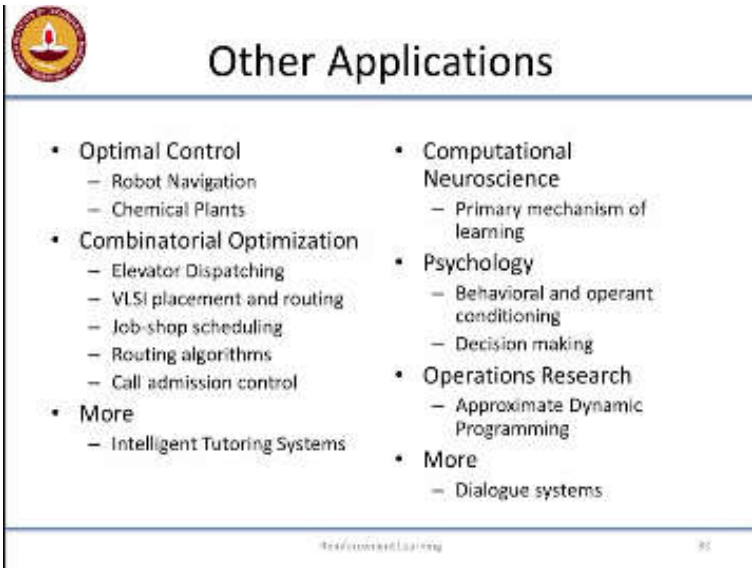
When I am doing the updating from my st + 1 is essentially the average accumulated over all the previous trends that I will be using , so if I play all the way to the end and update it will be with a 1 or a 0 but st + 1 could be anywhere between 1 and 0 depending on what is the probability of finding so I will be using that value for updation that is a crucial difference I so there are many different solution methods so there are all this which are called temporal difference methods so these are all different algorithms.

(Refer Slide Time: 05:03)



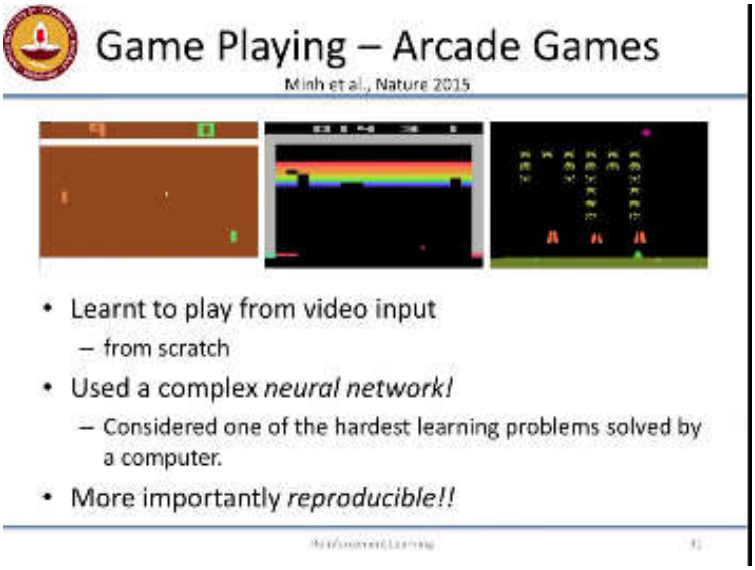
TD-λ, Q-learning, SARSA, actor critic so on so forth and then there is a soul search of algorithms which called policy search algorithms and then there is dynamic programming and their whole bunch of other applications for RL it so we could they are all over the place as you could see.

(Refer Slide Time: 05:23)



Optimal control optimization company too common a total optimization for psychology neuroscience so that is not a theory since I was asking is there anyone from biotech because biotech people do use reinforcement learning a lot and usually there are one or two people in the RL class so this is a surprise or maybe that is there was a bear trick with this according to the economic website maybe this is gave up in CS 36 and went back I do not know so here is the most recent.

(Refer Slide Time: 05:59)



The hot thing that comes from came from RL more game playing and for a change is not from IBM and it is from Google but the company that actually built the first this arcade game playing engine was called deep mind and as soon as deep mind built a successful engine Google bought them and so now it is Google deep mind but it is a separate entity it is not part of Google deep mind operate out of London and they bring all kinds of interesting stuff many of the hot advances very recent advances in the last year or so. In reinforcement learning seem to be coming out of deep mind so what they did was how many if you know about this Atari games , everyone knows about Atari games people are getting tired really no one has played pac-man yeah, how about how about the pong, break outs, space invaders come on yeah anyway so what happened was this is team in University of Alberta okay, which put out this their what they call the Atari, the arcade learning about the Atari learning environment on arcade learning environment which essentially they allowed computers to play these games .

These artery games and what the deep mine fellows came up with is a reinforcement learning agent that learn to play this game from scratch I just by looking at the screen okay that is all the input it was getting just the pixels from the screen they are all pixels on the screen but given as inputs to it used to very complex neural network so it is a deep learning deep network and it is considered of the hardest learning problem solved by a computer and I think I believe it is a one the only computational reinforcement learning paper ever appear in nature , so usually is very

hard for non natural science people to publish in nature and kind of obviously it is usually Hard for computer science to publish in nature but this was totaled out as a next step in trying to understand.

How humans process in post blah essence of all kinds of marketing jargon but more importantly than anything else about this it is reproducing them so I told you about I think that is a warning sign for me to stop so I told you about the helicopter so there is basically Stanford and Berkeley or the two people who get the helicopter to fly I told me about the backgammon player that is like Jerry Tessaro is to are is the one person who gets the backgammon player to work .

Partly because all the input features he uses in there or proprietary but partly because has a very hard problem to solve what is the amazing thing about this Atari game engine is that this case our release of code you can if you have enough powerful GPUs you can set it up here and get it to play and get a reasonably working engine that plays those places Atari games that is the amazing thing about it that is reproducible as opposed to many of the other things other success stories other success stories you had in the past so I do believe I had just one more slide after this so let us see if this will work.

(Refer Slide Time: 09:20)

Okay.

(Refer Slide Time: 09:24)

Oh so if you are really doubts the green one is the learning agent. No this just sped up for you to see you mean it is not like the game is progressing at the same rate but you can see the score slow Mind you it was not given a reward okay it is just given the screen never got to reward for winning ideas and understand that the pixels on the top or rewarding and if we give it rewards it becomes cheating yeah which is what they did they did they did add a game over which is misty the a heurists considered as cheating but they did not a game over sign so the longer you keep it going the better.

It is basically nothing this is getting boring so this is learnt here so this is a seaquest you have to swim and then sink down get some things and come up so seaquest is a game that it never learned to do greatly on. Seaquest is not something which did learn to solve well so there are a few games like this, so they initially published the nature paper I think they had liked like 45 or 46 out of the 50 atari games they were able to play well and I think in 43 of them it had better than human performance and I think the current state is they have like one game that does not play well and have better than human performance in like 48 of those.

Funded by
Department of Higher Education
Ministry of Human Resource Development
Government of India

www.nptel.ac.in

NPTEL

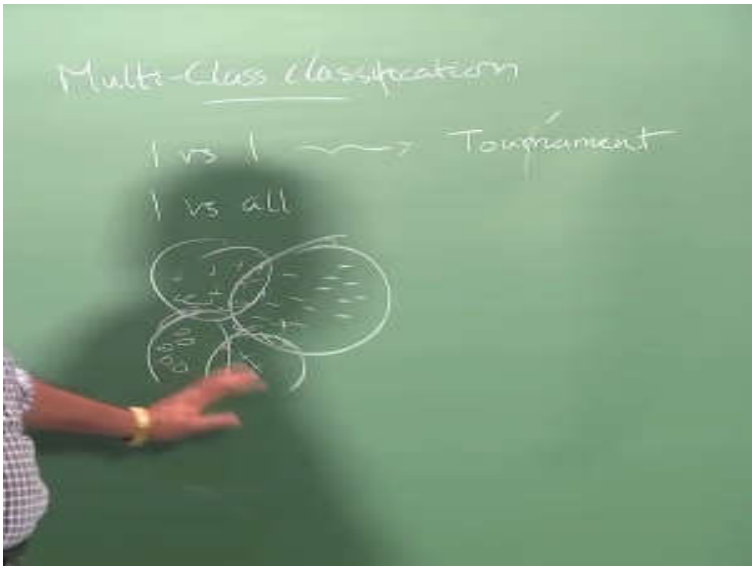
NPTEL ONLINE CERTIFICATION COURSE

Introduction to Machine Learning

Lecture-86
Multi-class Classification

Prof. Balaraman Ravindran
Computer Science and Engineering
Indian Institute of Technology Madras

(Refer Slide Time: 00:15)



There are some classifiers that we looked at which are naturally multi class classifiers , which are they you know neural networks yeah with a little bit of work here they are multi class something which is more immediately multi-class this decision trees is immediately multi-class no need to worry any fiddling around with anything else Naïve bayes and all the Bayesian classification that we looked at read all those are immediately multi class classifiers and then there are things which we looked at which are inherently two class classifiers .

SVM's is one of those every popular of those and any other two class classifiers that you know logistic regression is inherently a two class classifier but we have multi class variants of it I did the two class classifier in detail in class but there are multi class variants of logistic regression but again the two class one is the one that is best understood and any kind of discriminate function based classification that we looked at are inherently to class classifiers .

You mean you can think of ways of converting them into multi-class classification but inherently to class classifiers so suppose I give you a very powerful mechanism for constructing a binary classifier, can you solve the multi-class classification problem using that let us make it even more concrete I give you an SVM I will give you this packaged code for an SVM I am telling you this is the best possible SVM implementation but it does only binary classification can you use that and convert it into a multi-class classification think how do you do that? what is the advantage of one versus one? potentially be balanced . hopefully I mean depends so you still have an unbalanced class classification problem I might have 30 classes in which each class has 10 data points and one class has 10,000 .

So then one versus one will be a problem then but one versus all will always be a problem even if you have equal number of data points in every class one versus all will be a imbalance problem but the disadvantage of one versus one? how many classifiers you need in one versus one? n choose 2 , ? so that is a large number of classifiers I give you a hand 100 class classification problem so how many classifies will you need within 1 versus 1 large number .

So you are in the time and we actually want to the classified we need not run them all like at the time interval timer feel to the creation which would probably be a one-time process yeah we would need all okay and when you run when you actually want to classify how many do you need how many would be actually if I whenever you guys while you get rid of but one person any person do you how come because you for example if you have class A B C and D huh, I will run the classifier for a versus b a versus c a versus b then you run it for a versus B you through one of them out then suppose it was yeah b versus c and one come over here.

You could do that but then for that you have to be little bit careful because the guarantees that you would have or slightly weaker so this is really not called 1vs1 okay so that is one version of running 1 vs 1which is called a tournament is essentially what you are suggesting is run at tournament . So you train lot of 1vs1 classifiers and then you run a tournament so you need to train that many classifies but we are deploying it you will have fewer numbers to use .

But the problem with that is suppose you have A versus B classifier was weak then you would throw out A incorrectly and suppose we are running A versus B and then B won but if you have done here A versus C and A versus D also A might win against all of those and then B might lose to C and D then it becomes an issue so a would have gotten two votes well be would have got only one vote and then what do you do so classifies are good then tournaments or be great if classifiers are weak and you have problem in tournament that you might eliminate things a little early .

And then another problem in the tournament is you can identify only the most likely class but if I wanted to give ranking of class labels okay that cannot be done with tournaments okay but if you have hundreds of class labels , so you have some you have to give up on something so essentially give up on the correctness and they essentially try to run a tournament if you have a lot of class labels try running a tournament on this .

So the scikit- learn implemented tournament automatically you know in this one versus one yeah that is that is fine what about SVMs? It supports multi class SVMs but there is nothing called multi class SVM here we have to do one of these I will take it back that is a multi classifier we will take it back but there is not work scikit-learn yeah so but this might be something you might want to employ okay now going back.

So I told you that it is possible that we have severe class imbalance in a 100class classification problem even , so you have one class that has like a million data points and each of the other classes have like a thousand data points so what would you do in that case so we spoke about some ways of fixing the problem the class imbalance problem so weighing some one class more

than the other under sampling over sampling did we talk about this at some point I vaguely remember discussing class imbalance in the class, no we discuss class imbalance in the class .

So there are different ways of fixing that you could try that alternatively you could try some kind of a hierarchical classification , so what you do in hierarchical classification is you essentially try to split the classes into two groups okay. And then say that okay first level I will see whether it goes into group one or whether it goes to group to the next level or in within group one I will try to assign it to a specific class or I can split it in to groups three and four and then within that group three I will assign it to a specific class , you could do some kind of error K classification .

So what is the challenge here? So sorry choosing the hierarchies yes using the groups so unless the groups come to you somehow from the domain itself sometimes you could have like people classify web pages and then you can go and look into some open directory project or something like that and there are nicely classified web pages for you . So you have a hierarchy of web pages there you can then look at the classification down the hierarchy.

So you will start off with saying okay entertainment versus news and then within news you could have say politics sports and then within entertainment you will have movies and I do not know and sports comes into the news or entertainment. So wherever so you could I will have this kind of hierarchies and then you can use this hierarchy to give you your hierarchical classification in the absence of that how would you want to do this?

If you want to induce the hierarchy I give you a flat set of 100 labels if you want to induce a hierarchy on this hundred labels how would you do this? Hmm, based on the number of data points, clustering it is clustering so what do you do with clustering how do you do clustering in this case just cluster of the data points blindly that person you are essentially solving we do not know which way to solve it so how will you cluster it here people are throwing up all kinds of terms now .

So the point is you have, so the intuition is the following I have this class conditional densities I know okay this is class one what are the data upon this is class to what are the data points , so I

would like to group them in such a way that the class condition densities belonging to one group are very different from the class conditional densities belonging to the other group does that make sense .

Suppose my data is like this so I have all my class one here class two here class four here and class five data points are here four okay, so which what is what is a grouping that suggest to you itself suggests itself to you one and two should be one thing and three and four should be the other . So if you think about it so this is the class condition if we are assuming these are drawn from Gaussians and I will have a mean here and some variance over this and the mean will be somewhere here and some covariance oh these look closer then the means of these distributions.

So that is a basic idea one way of achieving this is to say that okay I will do clustering and then I look at the class labels that fall together which class labels fall together more often this is very nicely done . So what if my classes are actually like this the classes look like this now it is harder to harder to separate them out , so what you can possibly end up doing something like this and I find some clusters some groups of data points like that.

Now predominantly in this I have class one predominantly in this I have class two predominantly in this I have class three in predominantly this whatever prominently in this class three class four now I start looking at which is which of the clusters are similar and then I can do some kind of predict which values the training data is given to you . So you have the training data you are just being clustering on the training data.

So the training data will tell you what the class labels are , this is there is this really a formal way of doing it I am just giving you tips practical tips forgetting addressing some very large problems, sorry so I have done these clusters now I have clusters and I can I can figure out which clusters are similar to which cluster which clusters are close that I have some description like suppose I am using some kind of a Gaussian model for describing my clusters I will of some description of the clusters .

So I can I can now figure out which cluster is close to which cluster I will talk about hierarchical clustering depending on today or Friday, Friday stands for next class okay. So today or Friday I will talk about the hierarchical clustering then you can see that okay there are four clusters and then these two clusters get merged first and then these two clusters get merged so at that point I can say hey okay now I am going to say all the classes which are more prevalent in these two clusters should go together the classes that are more prevalent in these two clusters should go together.

The data points it belong to those clusters you go and build a classifier first classifier you build on all the data points that separates these two clusters from these two clusters that spacing so I do not want to do a distinction at the very beginning I do not want to do a distinction between this and that this and this and so on so forth . Then how does this help us in class imbalance?

Yeah what if you originally you had class imbalance what if originally 1 class had a million points and all the other classes had a thousand points each, yeah if that is supported if the data supports that million and this is extreme but say ten thousand to one class about the we do get real data like that. I did not get it, we were making clusters on the glass labels so it does not matter what the size of the cluster is so all points belonging to same level would fall into in the same cluster.

How it goes is class imbalance, no it will not cause imbalance I am asking how will it relieve you from class imbalance we are getting is less than has only see I am not I am not using the clustering itself to do the classification I am only doing the clustering turn that I can group the class labels and then I go back and try to solve the classification problem after that yeah. So I suppose all of you are going to try out different things.

IIT Madras Production

Funded by
Department of Higher Education
Ministry of Human Resource development