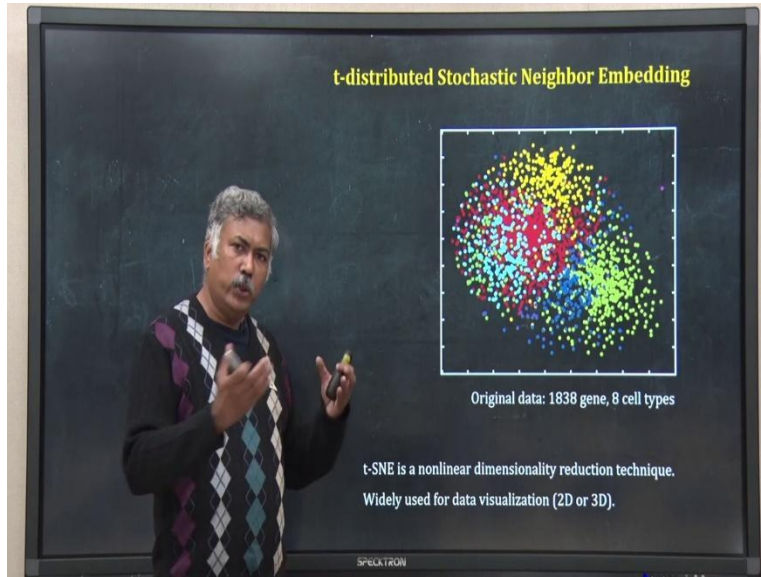


Data Analysis for Biologists
Professor Biplab Bose
Department of Biosciences and Bioengineering
Mehta Family School of Data Science and Artificial Intelligence
Indian Institute of Technology, Guwahati
Lecture – 47
t-SNE

(Refer Slide Time: 00:36)



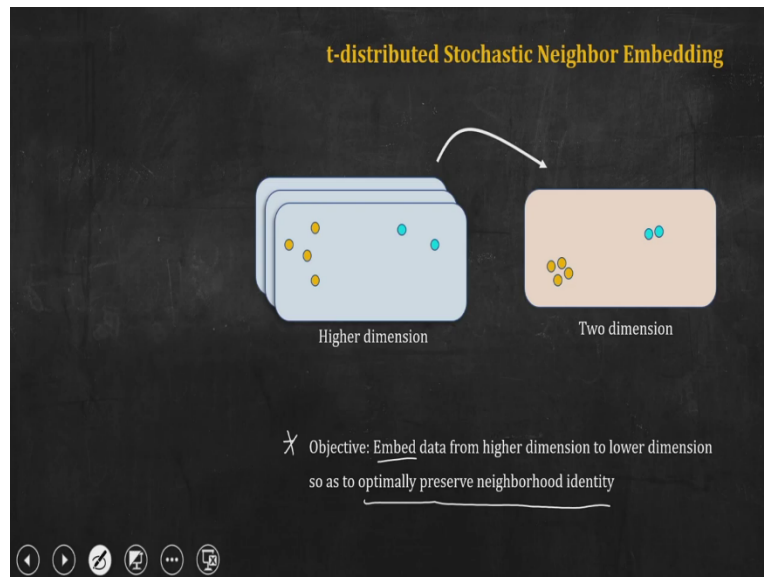
In this lecture, we will discuss t-SNE. t-SNE stands for t-distributed stochastic neighbor embedding. It is a quite a popular method to visualize higher dimensional data in lower dimension. And it is a nonlinear technique of dimension reduction. Now, before I go and explain what is t-SNE, let me give you one example. Here in this figure what I have done, I have a gene expression data set for around eight cell type, each cell type has hundreds of cells.

And for around 2000 genes, 1800 something we have measurement for gene expression. So, you can imagine the dimension of the data is then 18,000 something. So, we can visualize it. So, visualize, to visualize this data we have to reduce the dimension of it. So, what I have done I have used t-SNE to visualize or embed this data in two dimension. So, here is the two dimensional plot, each of these plot, each of these dot here is one cell.

And that belongs to one of the cell types and those are colored based upon which cell type they are. Now, here you can easily see the dispersion among the different cell types. There are some

amount of cluster in some case there is no cluster there is a diffusion also. So, this is how you can visualize your higher dimensional data like a gene expression data using t-SNE in two dimension or in three dimensions.

(Refer Slide Time: 02:07)



Now, unlike other dimension reduction techniques like PCA, t-SNE focuses on the issue of neighbor relationship. That means, it is tries to embed the data in such a way that the local information, local neighborhood information in the higher dimension is retained. So, the objective of t-SNE is in general is to embed the data from higher dimension to a lower dimension by optimally preserving the neighborhood identity.

Now, how that should be done? How should we keep a focus on the neighborhood and rather than on the global information, we focus on the neighbor information and try to embed by maintaining the neighborhood relationship, how can I do that?

(Refer Slide Time: 02:59)

Stochastic Neighbor Embedding

2D

d_{ij} : distance/dissimilarity between x_i and x_j

1. Calculate d_{ij} for all (i, j) data pairs
2. Calculate a probabilistic similarity measure using d_{ij}

Stochastic Neighbor Embedding

d_{ij} : distance/dissimilarity between x_i and x_j

Calculate d_{ij} for all (i, j) data pairs

Calculate a probabilistic similarity measure using d_{ij}

$$p_{j|i} = \frac{\exp\left(-\frac{d_{ij}^2}{2\sigma_i^2}\right)}{\sum_{k \neq i} \exp\left(-\frac{d_{ik}^2}{2\sigma_i^2}\right)}$$

$$p_{\frac{j}{i}} = \frac{\exp\exp\left(-\frac{d_{ij}^2}{2\sigma_i^2}\right)}{\sum_{k \neq i} \exp\exp\left(-\frac{d_{ik}^2}{2\sigma_i^2}\right)}$$

To understand how t-SNE work, we have to first understand the first version or first avatar of this method call SNE stochastic neighbor embedding. So, to explain SNE what I will do? I will take a two dimensional data, and then imagine I am trying to embed it in one dimension. So, from two dimension, I am going to one dimension. So, in this two dimension data suppose I have five data points and I have shown them here.

Now, for each of these data point, you take another point. So, you have a pair. For example, in the figure I have shown two yellow dot one is x_i and other one is x_j . So, use some measure of distance or dissimilarity between data points, for example, you can use Euclidean distance whatever is meaningful for you, then you calculate the distance between these x_i and x_j . And suppose that is d_{ij} .

In this way, I can have five choose two pairs right. So, I can have 10 pairs. So, I can calculate this distance between for each of these pairs, I should have d_{ij} for each of these 10 pairs. So, you calculate the pairwise distance data or dissimilarity data. But the interesting thing is in SNE, I will not use these distance data or dissimilarity data in higher dimension directly to embed in the lower dimension.

I will use some probabilistic measure that I will calculate from this distance measure. And how I will do it? I will explain that. So, what I am doing, I will calculate a probabilistic similarity from that d_{ij} . So, do that when you calculate a conditional probability. In this case, I am comparing x_i and x_j . So, the conditional probability p_j given i is given by this formula. Where in the numerator you have exponential to the power e to the power minus d_{ij}^2 , d_{ij} is the distance or dissimilarity between x_i and x_j .

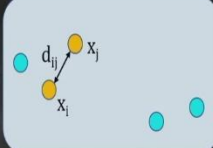
You are taking the square of that, divided by 2 into σ_i^2 . Essentially, σ_i is very instanst term, σ_i^2 is a very instant term. And I will explain that. In a numerator what you have? You have a summation thing, this part does the denominator part where you have the summation, this is actually nothing but normalization.

What you are doing here is that see, I have the calculated in a numerator I have calculated e to the power minus d_{ij}^2 by $2 \sigma_i^2$ for x_i and x_j pair. But now taking x_i you can pair up with other three data points. So, for all these four pair for x_i , you calculate this numerator value. And then you sum them up together and put that in that denominator, that will give us this p_j given i , conditional probability of j given i .

Now, this looks a bit complicated, is not it? Why are we doing this? Pay attention to this part, in this probability calculation is not very complicated. Essentially, this formulation this function is nothing but a normal distribution.

(Refer Slide Time: 06:23)

Stochastic Neighbor Embedding



d_{ij} : distance/dissimilarity between x_i and x_j

Calculate d_{ij} for all (i, j) data pairs

Calculate a probabilistic similarity measure using d_{ij}

$$p_{j|i} = \frac{\exp\left(-\frac{d_{ij}^2}{2\sigma_i^2}\right)}{\sum_{k \neq i} \exp\left(-\frac{d_{ik}^2}{2\sigma_i^2}\right)}$$

PDF of Normal distribution

$$f_X(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2}$$

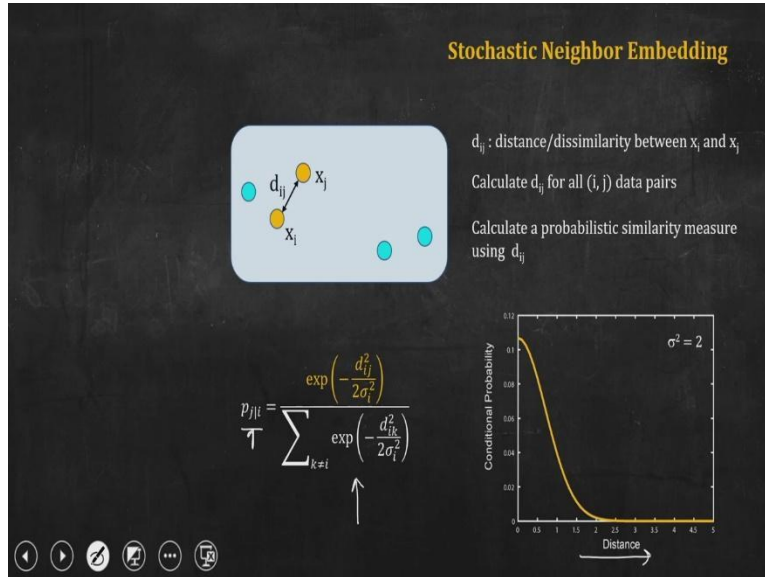
$$p_{j|i} = \frac{\exp\left(-\frac{d_{ij}^2}{2\sigma_i^2}\right)}{\sum_{k \neq i} \exp\left(-\frac{d_{ik}^2}{2\sigma_i^2}\right)}$$

$$f_X(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2}$$

Let me explain that this is the PDF of normal distribution, we have studied that earlier, very early in our course. And we focus here, in Gaussian distribution pdf, you have e to the power minus half x minus mu squared divided by sigma square. The equivalent thing is this one, in place of x minus mu, you have d ij.

And you have sigma here, you have sigma i here. And you have 2 here, and you also have 2 here. So, although this function looks a bit complicated, this p ji, p given i is nothing, this conditional probability is nothing but a Gaussian distribution. And what I did?

(Refer Slide Time: 07:13)



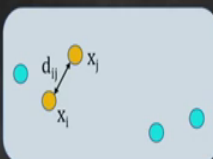
$$p_{j|i} = \frac{\exp\left(-\frac{d_{ij}^2}{2\sigma_i^2}\right)}{\sum_{k \neq i} \exp\left(-\frac{d_{ik}^2}{2\sigma_i^2}\right)}$$

I did just a simple plot, I set sigma squared equal to 2, this sigma equal squared equal to 2 and created a plot where I have in the horizontal axis I have a distance d_{ij} . And in the vertical axis, I have this conditional probability calculated using this formula. And the brown line is the result. You can easily see this is half of the normal distribution with a mean at 0. We do not have the other half because in my calculation, I have considered distance to be only positive, that distance has negative distance has no meaning.

So, I have taken distance 0 to up to suppose high higher value five, positive values five. So, I have half of the normal distribution. And you can easily see essentially nothing but Gaussian distribution. So, what we are doing, we are calculating this p_{ji} for x_i and x_j .

(Refer Slide Time: 08:10)

Stochastic Neighbor Embedding



d_{ij} : distance/dissimilarity between x_i and x_j

Calculate d_{ij} for all (i, j) data pairs

Calculate a probabilistic similarity measure using d_{ij}

$$p_{j|i} = \frac{\exp\left(-\frac{d_{ij}^2}{2\sigma_i^2}\right)}{\sum_{k \neq i} \exp\left(-\frac{d_{ik}^2}{2\sigma_i^2}\right)}$$

σ_i is controlled by a tuning parameter called **Perplexity**.

$p_{ij} = \frac{p_{j|i} + p_{i|j}}{2N}$

Total data

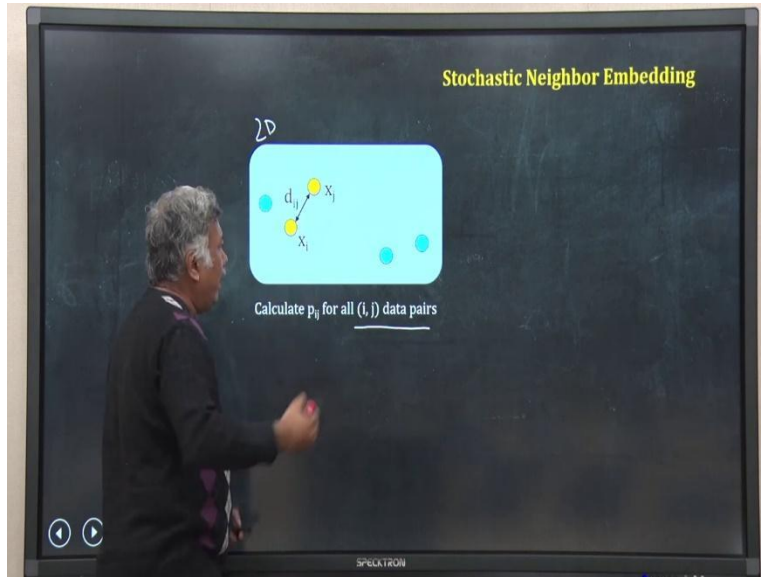
$$p_{j|i} = \frac{\exp\left(-\frac{d_{ij}^2}{2\sigma_i^2}\right)}{\sum_{k \neq i} \exp\left(-\frac{d_{ik}^2}{2\sigma_i^2}\right)}$$

$$f_X(x) = \frac{1}{\sigma \sqrt{2\pi}} e^{-\frac{1}{2} \left(\frac{x-\mu}{\sigma}\right)^2}$$

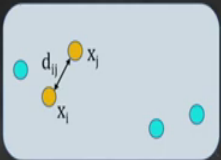
Similarly, I can also calculate $p_{i|j}$ given j conditional probability of i given j , using the same formula. Now, you sum them together and you divide by 2 into n , where n is the total number of total data, total number of data points and I get a joint probability p_{ij} . So, I will in my calculation, now, I will forget about p_{ij} , the distance or dissimilarity between x_i and x_j .

Rather, I will do all further calculation using this p_{ij} . And while doing this calculation, if you have noticed that I have to set this value, σ_i . Because in this equation p_{ij} is coming from the original data set the dissimilarity data that I have in higher dimension. But how do I know σ_i ? σ_i is this is actually a parameter in your this algorithm. And that is decided by a tuning parameter that you can tune as a user and that is called perplexity. So, once you set the value of that perplexity, the σ 's value will also be decided.

(Refer Slide Time: 09:27)



Stochastic Neighbor Embedding

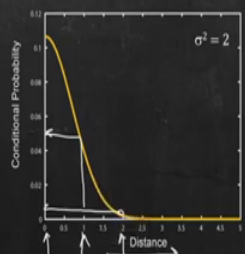


d_{ij} : distance/dissimilarity between x_i and x_j

Calculate d_{ij} for all (i, j) data pairs

Calculate a probabilistic similarity measure using d_{ij}

$$p_{j|i} = \frac{\exp\left(-\frac{d_{ij}^2}{2\sigma_i^2}\right)}{\sum_{k \neq i} \exp\left(-\frac{d_{ik}^2}{2\sigma_i^2}\right)}$$



$$p_{j_i} = \frac{\exp\left(-\frac{d_{ij}^2}{2\sigma_i^2}\right)}{\sum_{k \neq i} \exp\left(-\frac{d_{ik}^2}{2\sigma_i^2}\right)}$$

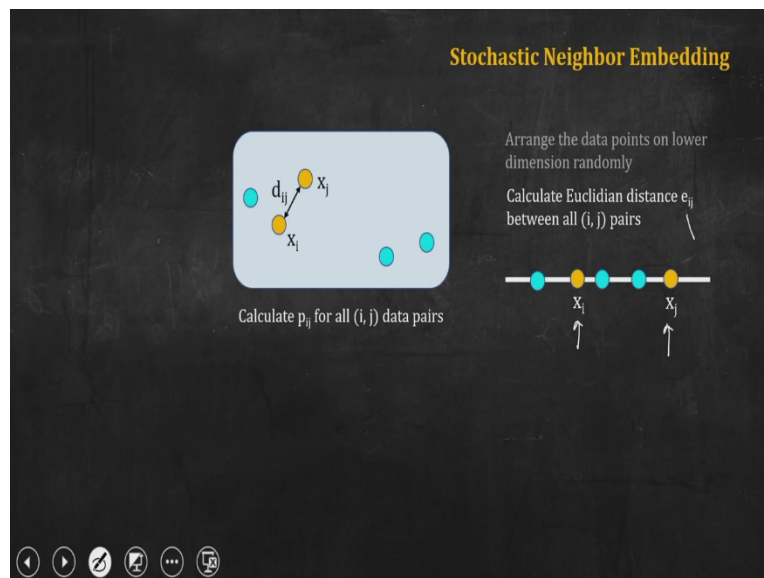
So, in this way, I can calculate p_{ij} . For all data pairs, in this case, I have five data points that means, I can ten pairs. So, I calculate p_{ij} , the joint probability from that distance measure. I can tell I calculate all these 10 probability values. Now, as I said in place of distance between these 10 pair, in these, for these 10 pairs, I will use their corresponding probabilities. Before I move further, let me explain another issue here.

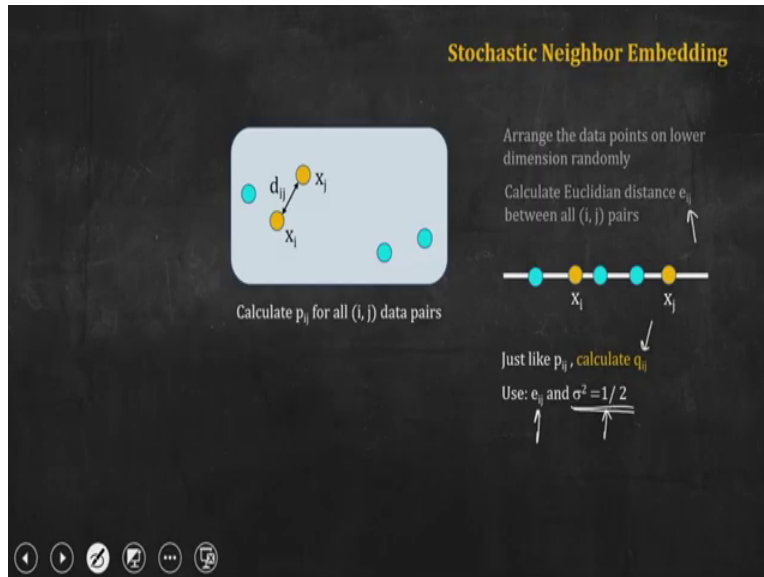
See, I have say I am saying that I will not use directly the distance d_{ij} . But I will use the probability p_{ij} . So, how will that get changed? How the p_{ij} will affect my further calculation? See suppose, if I take one data suppose a distance is on 1, suppose the distance is 1. So, then if I can say a d_{ij} equal to 1, then the corresponding probability somewhere is here, suppose 0.05. Now, let me take a double distance suppose the distance between x_i and, x_i and x_j is 2.

So, I have doubled the distance, what is the probability? Probability is in very low value. So, you can see, if I have used the original distance major d_{ij} then I am just doubling the distance, I am just doubling the dissimilarity. But, in if I use this normal distribution, Gaussian distribution to transform that distance to probability term, then the dissimilarity score that weightage will become very large.

Because you can easily see it has fall such an amount. So, that is the advantage of using these normal distribution rather than calculating the probability and rather than using the original distance or dissimilarities. Now, let us move forward, this is the way I have calculated all the probability all these 10 probabilities. Now, I have to embed these data using these p_{ij} s to some way, in some way on a lower dimension. So, my system is two dimension as I said.

(Refer Slide Time: 11:44)





So, I will embed it in one dimension. So, one dimension will be a line. So, what I have done, the first step of the algorithm will do, it will arrange the data point in this lower dimension that means, in this line in this case randomly, without any bias. Suppose, it is randomly distributed. Now, what algorithm we will do, it will calculate Euclidean distance for each pair.

So, I have 10 pair for example, now, if I take x_i and x_j , then it will calculate the Euclidean distance in this lower dimension, while it has randomly placed all these data points. So, it will calculate e_{ij} the Euclidean distance. So, in this way it can calculate the e_{ij} Euclidean distance for all these pairs possible, in this case 10 pairs. Now, as we have now are not using the distance measure in higher dimension directly, we are calculating a probability based on the Gaussian distribution using the distance measure.

In this case, also, we will use a same normal distribution, same function, but what we will do, we will use e_{ij} in place of d_{ij} . E_{ij} is what? E_{ij} is the Euclidean distance of this data, now, arranged in the lower dimension. In this case, one dimension and I will do some other change also. We will consider the sigma squared the variance as constant. It will not affect much of our calculation, but it will make life easy.

So, we will consider this half. If you put this value in the original equation that we discussed a few slides back, you will find it will cancel 2 in the denominator in the exponential term. So, my calculation will be much easy and it will be uniform. So, what I am doing just like p_{ij} , I am

calculating q_{ij} . But now for the lower dimensional data, where I have placed the data randomly, but I am using the same formulation, same normal distribution or Gaussian function.

And I am using e^{-ij} and I am using sigma squared equal to half. So, I have 10 pairs in this example that means all the pairs possible.

(Refer Slide Time: 13:55)

Stochastic Neighbor Embedding

Calculate p_{ij} for all (i, j) data pairs

Calculate q_{ij} for all (i, j) data pairs

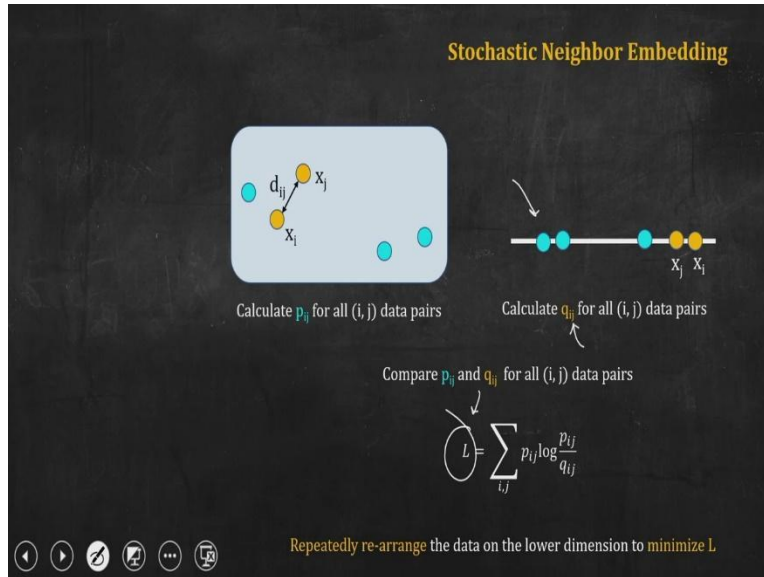
Stochastic Neighbor Embedding

Calculate p_{ij} for all (i, j) data pairs

Calculate q_{ij} for all (i, j) data pairs

Compare p_{ij} and q_{ij} for all (i, j) data pairs

$$L = \sum_{i,j} p_{ij} \log \frac{p_{ij}}{q_{ij}}$$



$$L = \sum_{i,j} p_{ij} \log \left(\frac{p_{ij}}{q_{ij}} \right)$$

So, I calculate q_{ij} for all of those pairs. So, now, what do I have? For higher dimensional original data set I have at least a p_{ij} , in this pristine example 10 pairs are possible. So, I have 10 p_{ij} values. Similarly, in lower dimension where I have randomly placed the data points, I have q_{ij} . Now, if I have arranged the data correctly in the lower dimension, then what do you expect? Then you will expect that the neighborhood relationship that means the dissimilarity or distance between each of the data point is retained.

And as I am using both the same function to calculate the probability, then the p_{ij} calculated for higher dimension and q_{ij} calculated for lower dimension they should be exactly the same, is not it? So, if I make a table a p_{ij} value and q_{ij} value, and if I have maintained these neighborhood relationship, the dissimilarity between data points in higher dimension, exactly same in lower dimension, then these p_{ij} and q_{ij} should be same.

But obviously, you know that these can, it is not possible because anyway, I am taking the data from a very high, higher dimension to a very low dimension. But your intention, the algorithm should try to keep p_{ij} and q_{ij} close to each other for all the pairs. So, that is what we have to focus on now. Now, what we will do?

We will use a loss function or cost function to calculate the difference between these p_{ij} values and the q_{ij} values. And the loss function or cost function that will be used is called KL divergence, we will not go in details of that. But intuitively, you can understand what this equation is telling. See, what I have here inside this summation, I have \log of p_{ij} divided by q_{ij} . Now, remember, my grand goal is that the ideal situation would be that if all q_{ij} is same as all the p_{ij} values.

Now, if that is true, if somehow I can manage to do that, then what will happen? This p_{ij} and q_{ij} here will be same. So, they will cancel each other. That means, I will get \log one and I will get \log one for all the pairs of data. Now, \log one is 0 and you are doing a summation. So, you will have summation of these 0s. So, the loss function will be 0. But that is a ideal case, you will not get a situation where in reality that loss function will be 0.

But as you can understand here, this loss function, the way we have defined, when my embedding of data from higher dimension and lower dimension is perfect in that case, I will take the minimum value and that minimum value will be 0. So, now, we have got a handle to optimize, use some optimization technique to solve this problem.

I started by randomly embedding, randomly placing the data from two dimension to one dimension, from a higher dimension to lower dimension. And I calculated q_{ij} then I calculated the l , the loss or the cost. Now, what I will do my algorithm will rearrange the data again in the lower dimension here in this line, in such a way that l will get minimized. And it will keep on doing that iteratively. So, the objective now is to minimize the l .

And to do that it will keep on jumbling rearranging the data in this lower dimension, keep on calculating q_{ij} , you calculate the l and once it has minimize it will stop. So, that is my final result, that is my final embedding, the best embedding I could have achieved using this iterative optimization method. That is what is stochastic neighbor embedding.

You are not using directly that distance or dissimilarity measure in higher dimension, you are using that dissimilarity score or distance measure of higher dimension in pairwise data to convert them in some probability, joint probability using Gaussian distribution. Because in Gaussian

distribution, what we are gaining, getting that the, the distance or dissimilarity is big, the weightage or score become lower.

Whereas when they are close, that means they are close neighbor, they are weightage becomes very high. Using that, I calculate p_{ij} . And then I rearranged the, I put the data or embed the data into a dimension in such a way that when I will calculate q_{ij} there, using the same formula, the difference between p_{ij} and q_{ij} is the least one. And the embedding, the distribution of the data and lower dimension that gives me the least cost, least I value is my final result. That is all SNE.

(Refer Slide Time: 19:23)

t-distributed Stochastic Neighbor Embedding

In tSNE use t-distribution in place of Normal distribution
to calculate q_{ij} for embedded data

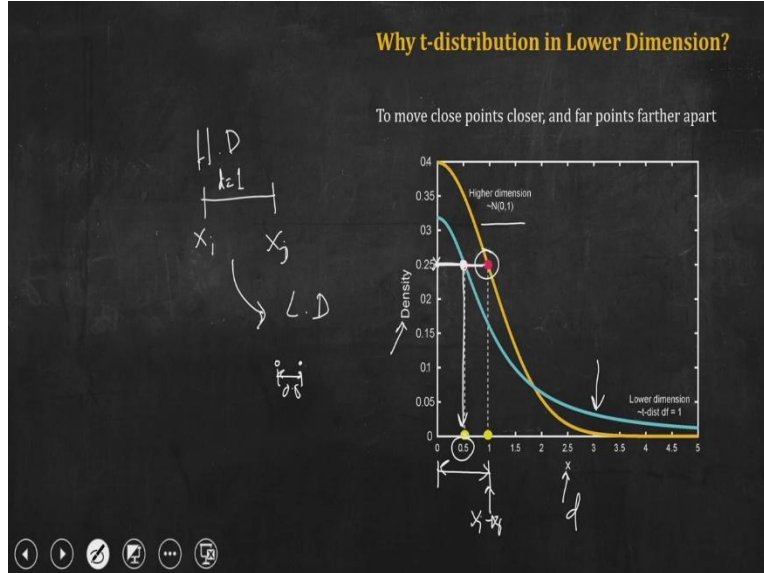
$$q_{ij} = \frac{(1 + e_{ij}^2)^{-1}}{\sum_{k \neq i} (1 + e_{ki}^2)^{-1}}$$

$$q_{ij} = \frac{(1 + e_{ij}^2)^{-1}}{\sum_{k \neq i} (1 + e_{ki}^2)^{-1}}$$

Now, we will move into the next up SNE, that is called t-SNE. T-SNE will not use normal distribution, remember in SNE I am using Gaussian or normal distribution both for the higher dimensional data as well as for calculation of q_{ij} in the lower dimension also. But in t-SNE, we will not use Gaussian distribution for calculation in lower dimension rather it will use t distribution to calculate this q_{ij} in the lower dimension where I have embedded the data.

And I have given the function by which they actually calculate the q_{ij} . Now, the key question here is why I am ditching normal distribution in lower dimension and going for t distribution?

(Refer Slide Time: 20:16)



That I will explain. The objective here is that see, I want to focus more on local information in t-SNE, not on the global information. So, you want to perform the embedding in such a way that the points which are close in higher dimension become much closer or compact in lower dimension. Whereas data points which are far away in higher dimension also move away, repel each other in the lower dimension, that you want to do, that you want to intentionally do in t-SNE.

So, you will, you want data points which are closer neighbor in higher dimension, you want them push them together, close together in lower dimension. Whereas data which are far away in higher dimension, you want to force them away. And that is nicely achieved using t distribution. Let me explain, what I have shown here. So, I have x you can consider these equivalent to distance or dissimilarity and you have density which are the probabilities we were calculating earlier.

I have shown two distribution, obviously, they are half because I am not taking negative values for x or d. So, this yellow one is a normal distribution with 0 mean and one variance. And you can imagine this I only use in the higher dimensions. So, they represent the probabilistic calculation in the higher dimension. Whereas this blue line is a t distribution of degree of freedom one, and it can be used for lower dimension.

So, the blue yellow line here, curve will be used for calculation in the higher dimension, for p_{ij} calculation. And suppose the lower one dimension q_{ij} calculation, we will be using this t distribution. Now, suppose in higher dimension, the distance between a particular pair, a particular pair is 1. Suppose that pair is x_i and x_j . So, the distance between or dissimilarity score between them is 1.

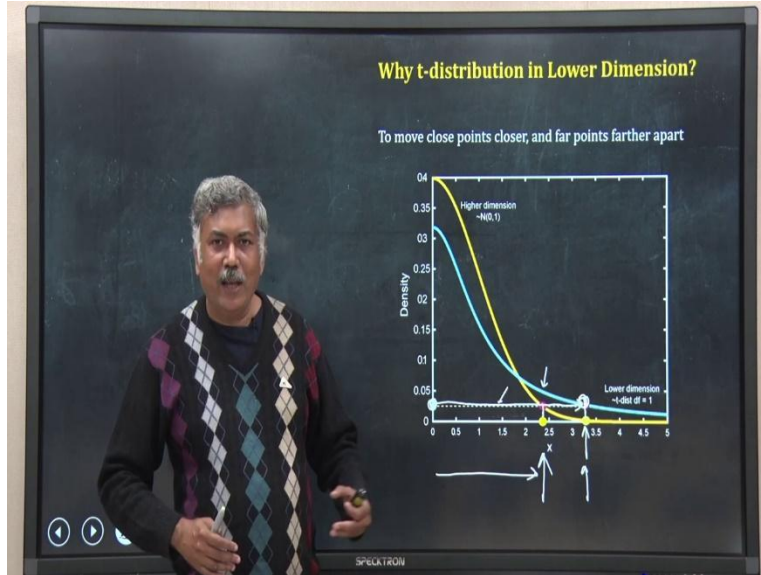
So if I calculate the probability using a Gaussian distribution, I will get a score something like this one, 0.25. Now, this is p_{ij} , now, what do you want? You want to embed these two data points in lower dimension in such a way that you when you calculate q_{ij} , the probability in lower dimension that value is very close to this value. And the perfect embedding will be when they are same.

So, consider that you have done achieve perfect embedding. That means, in the lower dimension, also, the q_{ij} value or the probability value will be this 0.25. Fine, I want to achieve that in the lower dimension. But in lower dimension in t-SNE, I am not using the Gaussian distribution in calculation anymore, I am using t distribution. So, then I move from here and I land up in these t distribution curve and go down on the distance axis.

So, that means in the lower dimension, I have to put the data with a distance 0.5. So, that means if in higher dimension, in higher dimension, if x_i and x_j were at a distance of 1, when I will embed that in one dimension, and if I am using t-SNE, the best one would be if I put them at a distance of 0.5.

So, that means, these two data points as they were close in the higher dimension, they become much closer in lower dimension. One part of my objective is done. Now, let us look about those data points which are far away in higher dimension, what will happen.

(Refer Slide Time: 24:24)



So, imagine now that the distance between x_i and x_j is this near 0.25. So, I will not use it directly that value, I will calculate the probability. So, I will go up in this Gaussian curve and I will calculate the probability along these horizontal dotted line, and this is the probability. Now, I want to embed the data in lower dimension. In ideal case, in the lower dimension also the probability q_{ij} should be same as that one.

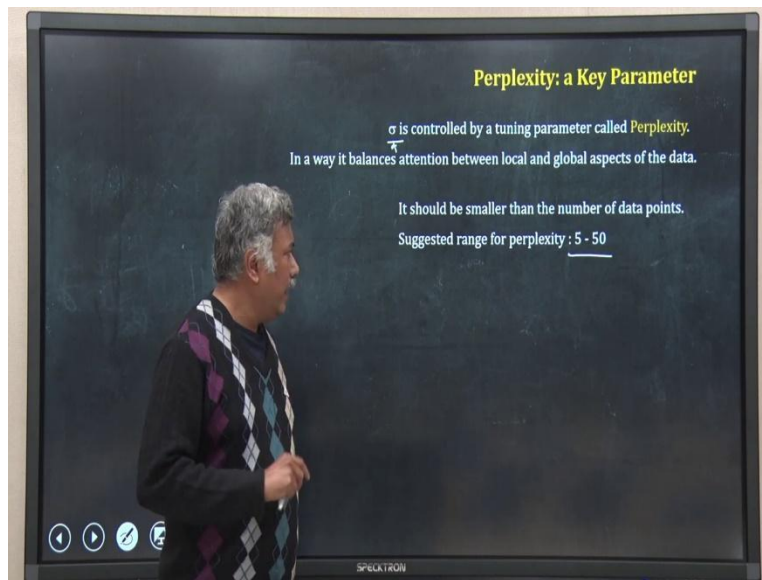
Let us move along this line horizontally. Now, for lower dimension I am using t distribution this blue line. And if you remember t distribution for less degree or lower degree of freedom has a fat tail and that differentiate it with respect to normal distribution, there is a fat tail. So, as you can see the tail near that position is much fatter in t distribution in the blue curve than the yellow brown curve, which is normal distribution.

So, now, I have reached this point which is on the t distribution curve. And if I look into the distance corresponding distance this and the distance is this one. So, in higher dimension the distance is around 2.5. But when you will embed that using t distribution, it will, the distance between these two points will become something near 3.5. So, these data points have shifted away, they have repelled each other.

So, my objective is achieved. By using t distribution in the calculation for lower dimension embedding t-SNE achieve as this, that it compacts the data points which are closer in

neighborhood in higher dimension much more close. And it pushes away data points which are farther away in the higher dimension, the it pushes is farther in the lower dimension. That is all how is SNE and t-SNE works and the differ.

(Refer Slide Time: 26:25)

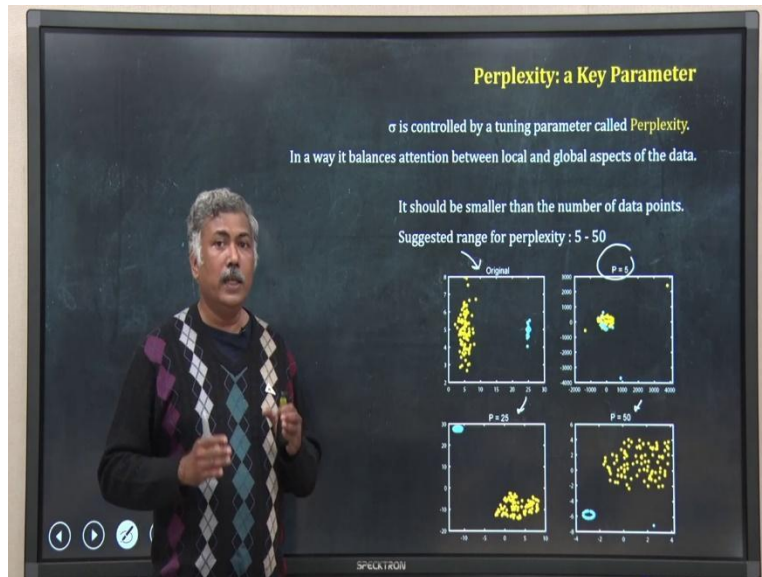


Now one important point, as I said earlier, that the sigma, the variance in my probability calculation in higher dimension is controlled by a tuning parameter called perplexity. I will not going details of the mathematical formulation of perplexity, but let us just to get a feel of it what it is doing, it essentially decides how much attention you put on the neighborhood rather than the global features, global distribution.

So, it decides how much in a way decide, how what should be the neighborhood. Whether neighbor should be big or the neighbor should be compact something like that. And you should never take a perplexity value, which should not be ever be small, smaller than the number of data points. And people have suggested that the range of perplexity values should be between 5 to 50.

But although these are thumb rules, you have to be very careful when you are performing t-SNE. Because changing perplexity values, the p value, in this case, if I write shortly, can actually change the whole embedding. And actually, it can spoil the whole, your whole purpose. Let me give you one example.

(Refer Slide Time: 27:39)



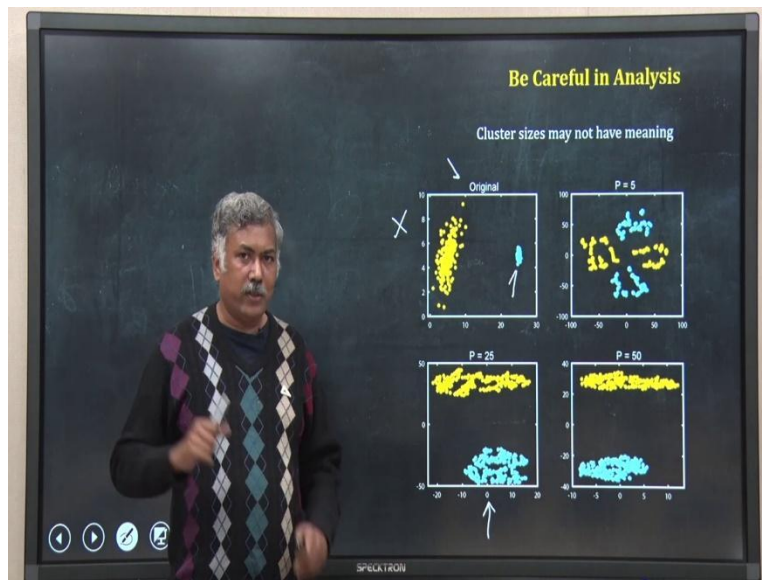
What I have? This original one, what I am doing actually, I am using a synthetic data. So, this is the original one. So, we have a two dimensional synthetic data. And you can see it has two groups, yellow one and the brown one. And I will embed using t-SNE in the two dimension itself. So, that you can easily understand what is happening because I, we will change the perplexity value in each of these case.

So, I have tested three perplexity value 5, 25 and 50. You can see when p is low all these groups, the green, yellow and the blue one, they are clumped together. Although in original space, original data, they are separate groups, but now they have come together. Now, what has happened here is that as you have considered perplexity value very low, it has focused too much on the local information.

So, it has not bothered to focus that okay, I had globally two groups, it has considered all of them as a single unit and focused on narrowly on around the neighborhood of each of these data points. So, it has collapsed everything in one club, one cluster. If I consider p equal to 25, I still have two groups you can easily see. Now, if you use p 50, then strange shapes and large dispersion comes.

So, that is why you cannot actually use any particular fixed value or perplexity for your particular work, you have to may have to try multiple value of p perplexity and find out which one is creating a meaningful embedding.

(Refer Slide Time: 29:20)



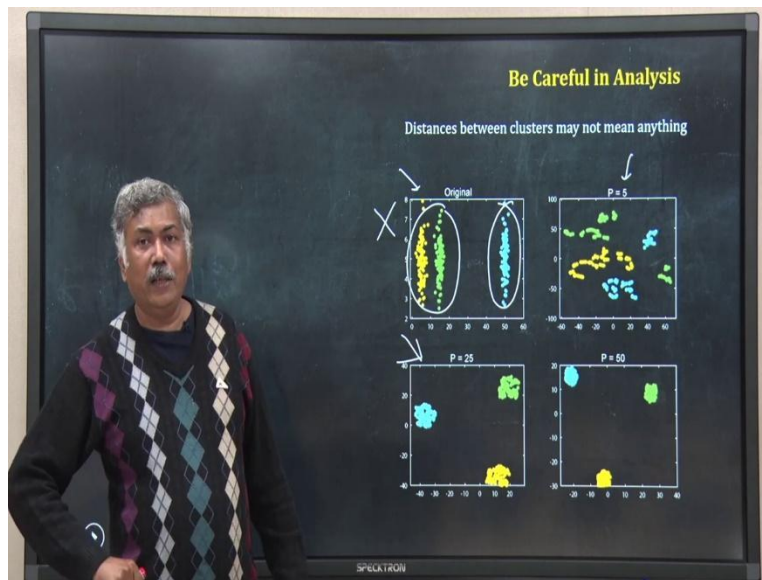
Now, I will discuss two important issue in t-SNE. So, t-SNE is a very good and popular method for visualization. You want to embed the higher dimensional data in lower dimension and visualize it. But sometimes, by mistake we try to take extract different meaning out of this visualization. And we should be very careful about that and we should avoid that. For example, what I am trying to show here again with synthetic data, these are original.

I have two groups blue and yellow, and they are segregated in two dimension properly. And I have done t-SNE with different perplexity value 5, 25 and 50. And you can see obviously, the results are different. Now, in this case, when you will do in real life, you will not know the original one, you will not know that. In this case it is simulated data so, I know it. So, suppose you have done with t equal to 25.

And you have seen that a both the blue group and the brown group are quite large group and they are large clusters. So, you may tend to conclude that both these groups are large cluster, I am talking of the size not the number of data points, as whole size, the area if you consider. But in reality actually that is not true.

In reality, the blue one is very compact, very compact, whereas, the yellow one is quite dispersed. So, just looking at that t-SNE plot, we should not try to make a meaning about the compactness or the cluster size in the higher dimensional data, we should never try to make a meaning out of it.

(Refer Slide Time: 31:07)



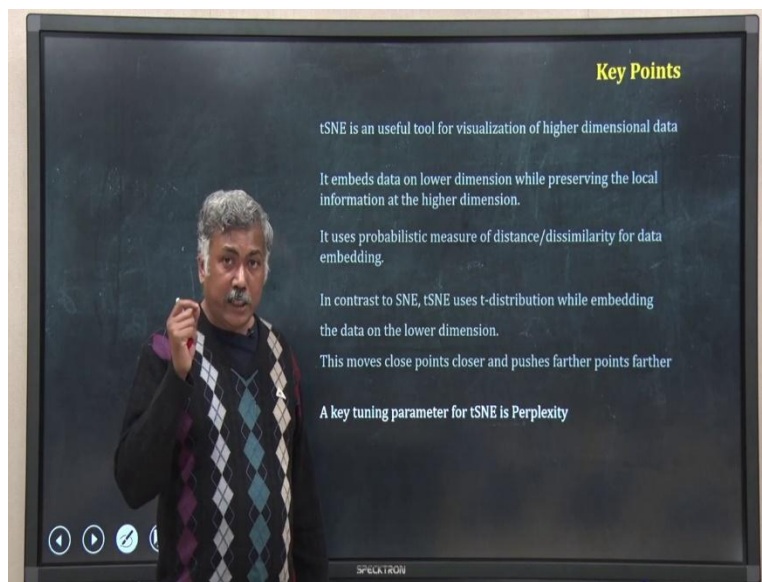
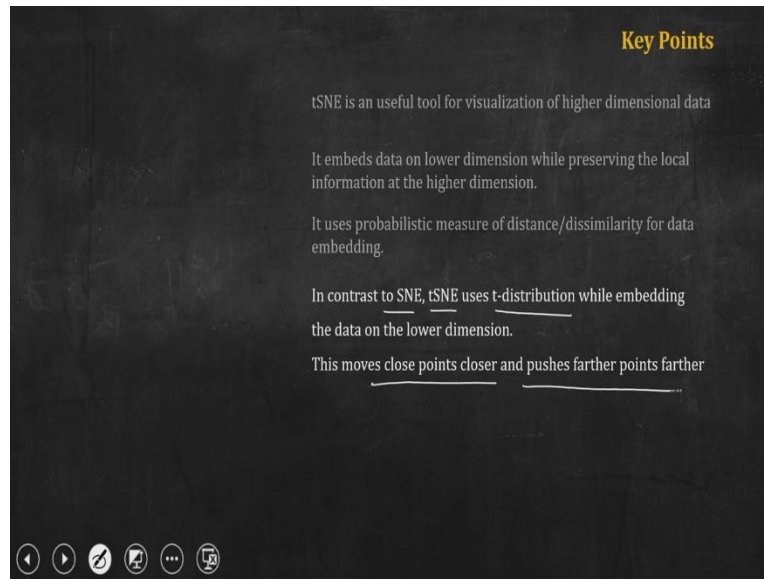
Next one, again a common trap we usually fall into. This is the original data again simulated. So, I have three groups in three different colors and I have performed t-SNE with three perplexity values. This p equal to 5 is very low, forget about that, consider 25 and 50. In all these cases, you can see I have three clusters and they are color coded properly. So, that there is no mixing of groups, and they are almost equally distance to each other.

Now, remember when you will do the real life t-SNE you will not know these data. It is simulated data that is why I know the original one, in real that I do not know the high dimensional data how it is dispersed. You will only see these ones, suppose you have done this. So, once you have gotten this plot you may tend to conclude that the in the original data these three groups are well dispersed, they are well away from each other.

But that is not true, look into the original data, these yellow group and green group are close to each other than this blue group. So, how they are separated? What is the distance between different cluster in my t-SNE plot in two dimension may not match with how they are dispersed

or organized in higher dimension. So, we should not try to make a meaning of the distance in the cluster, in our t-SNE plot. That, this brings me to the end of this lecture.

(Refer Slide Time: 32:36)



Let me jot down the key point. So, t-SNE is a very useful tool for visualization of higher dimensional data in two dimension or in three dimensions. And it is quite often used for that purpose. It embeds the higher dimension data in one dimension, while preserving the local information or the neighborhood information or neighborhood relationships in the higher dimension.

Now I uses a probabilistic measure, as we said in ordinal SNE it will use the both Gaussian distribution for both the calculation and in t-SNE it will use Gaussian and t distribution. So, it is a probabilistic measure of distance (())(33:14) dissimilarity between data points or pairs. And as I said, in contrast to SNE, t-SNE use t distribution.

The advantage of using t distribution as we discussed earlier is that it will put, close point closer and it will push farther points away from each other. Now one key parameter while you are performing t-SNE is perplexity. And as I explained with diagram, you should choose this perplexity value very judiciously. That is all thank you for learning with me today.