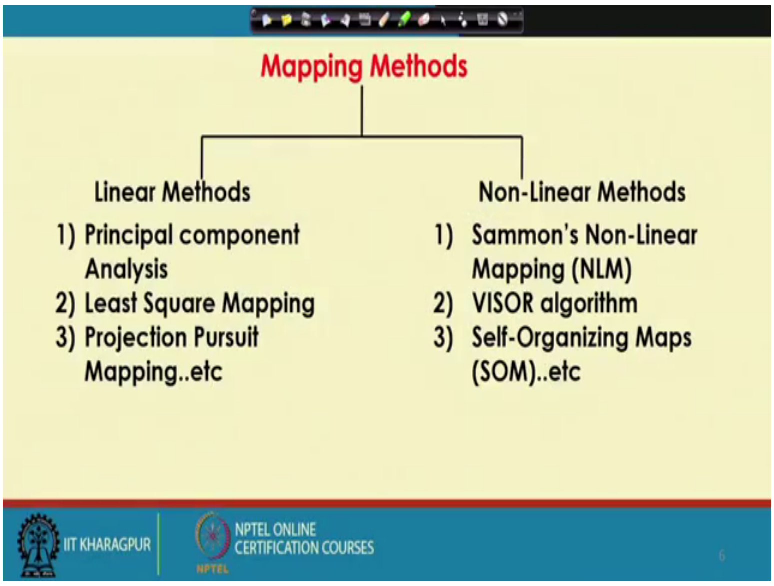
Traditional and Non-Traditional Optimization Tools
Prof. D. K. Pratihar
Department of Mechanical Engineering
Indian Institute of Technology, Kharagpur

Lecture - 14
Faster Genetic Algorithms (Contd.)

Before we start the principle of a faster genetic algorithm namely Visualized Interactive GA, VIGA. Let us explain the working principle of some non-linear mapping tools which we generally use to map the higher dimensional data to lower dimension for the purpose of visualization. Now, visualization is required just to find out the information of the surface of the objective function during optimization. If we want to find out the most appropriate search direction, now by knowing the most appropriate search direction we can accelerate the search of the GA just to make it faster.

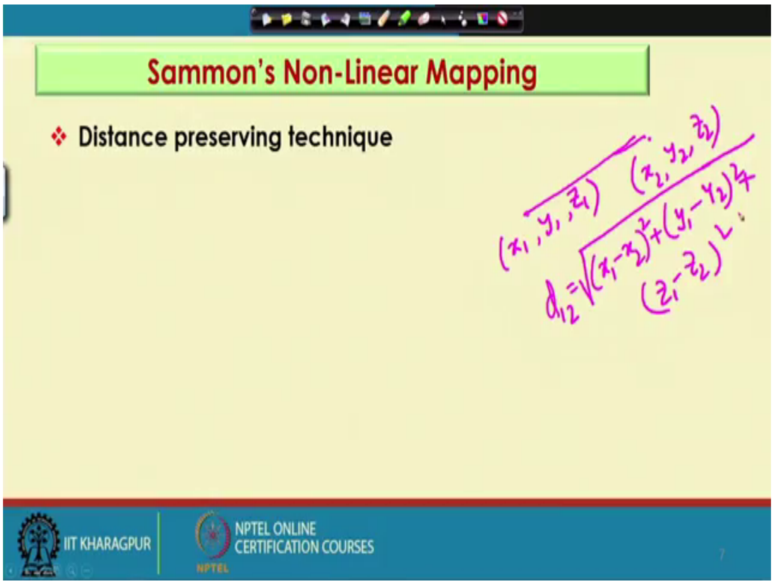
Now, today I am just going to discuss in details the working principle of the some of the very popular non-linear mapping tools.

(Refer Slide Time: 01:30)



Now, let me start with the first method that is Sammon's non-linear mapping.

(Refer Slide Time: 01:33)



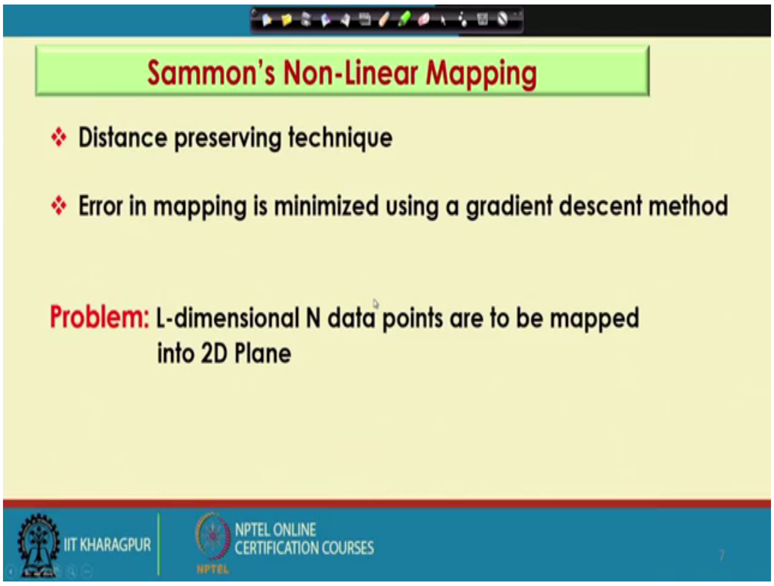
Now, here actually what we do is we try to map the data from higher dimension to lower dimension approximately and this is non-linear because from higher dimension to lower dimension exact mapping that is one is to one in mapping is not possible.

Now, the Sammon's non-linear mapping is a distance preserving technique. Now, let me explain what do you mean by the distance preserving technique. Now, let us take the example. Now, the tip of my finger supposing this is the first point and another tip of my finger is the second point, now I want to determine the Euclidean distance between the first point and the second point. Now, for the first point the coordinate can be represented as follows for the first point it is x 1, y 1, z 1 in 3D space and for the second point the coordinate is x 2, y 2, z 2. So, what you do is we try to find out the Euclidean distance between these two points and the Euclidean distance between one and 2 is nothing, but the square root of x 1 minus x 2 square plus y 1 minus y 2 square plus z 1 minus z 2 square. So, this is the way we try to find out the Euclidean distance.

Now, supposing that we have got the Euclidean distance values between the points 1 and 2. Now, once you have got this particular Euclidean distance we can keep it the same, but I can move the second point with respect to the first point by keeping the same Euclidean distance. For example, say the point one is fixed now point 2 I am moving keeping the same Euclidean distance. Now, if this is the situation on changing the topology of the second point with respect to the first point although I am keeping the same Euclidean distance. Now, here in this particular technique what we do is we try to keep the Euclidean distance between the 2 points constant, but we do not take care of its relative position of the second point with respect to the first point and that is why this is a distance preserving technique not the topology preserving technique.

So, this particular technique is a distance preserving technique. Now, let us see how does it work.

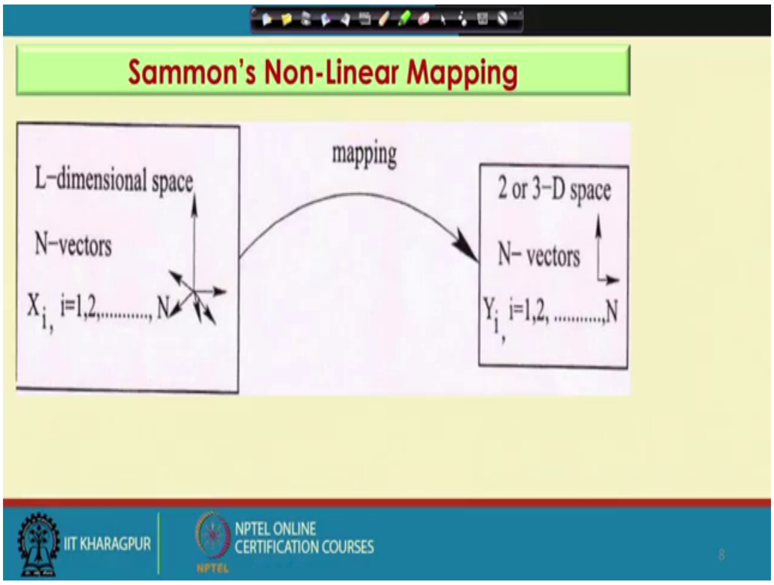
(Refer Slide Time: 04:59)



Now, we try to find out the error in mapping while mapping the higher dimensional data to the lower dimension; that means, supposing that I have got a large number data points in higher dimension say one thousand data points. I will have to find out one thousand corresponding data point either in 2D plane or in 3D space. Now, while doing this particular mapping, I will have to do the mapping in such a way so that the error in mapping becomes the minimum. Now, this is a minimization problem and to solve these we use a gradient based method in Sammon's non-linear mapping.

Now, supposing that I have got a problem involving capital N number of data points and each data point is having capital L-dimensions. Now, this data points are to mapped to the 2D plane.

(Refer Slide Time: 06:04)

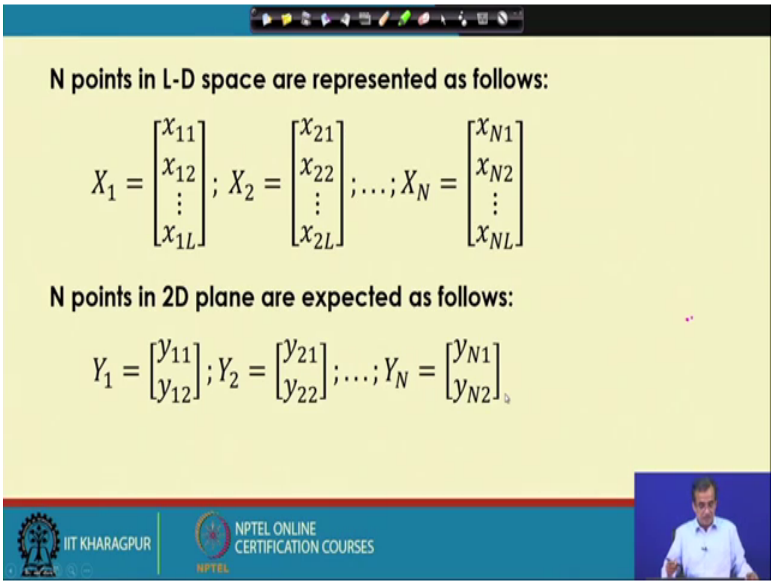


Now, this shows actually the schematic view of this problem which we are going to tackle. So, here we can see that we have got N number of data points in L dimensional space and this data points are represented by X i, where i is 1, 2 up to capital N and this is nothing, but the representation of L dimensional space.

Now, here in this particular L D space if I want to represent a particular point. So, I will have to use capital L number of numerical values. Now, this data points are to be mapped to either 2D plane or 3D space. Now, here for simplicity let me consider that I am just going to do the mapping onto a 2D plane. Now, it has got X and Y only 2 dimensions and once again. So, I will have to get n data points here and each data point should have only 2 dimension.

Now, here n data points are represented by capital Y i, where i varies from 1, 2 up to n. Now, let us see how to do this particular the mapping.

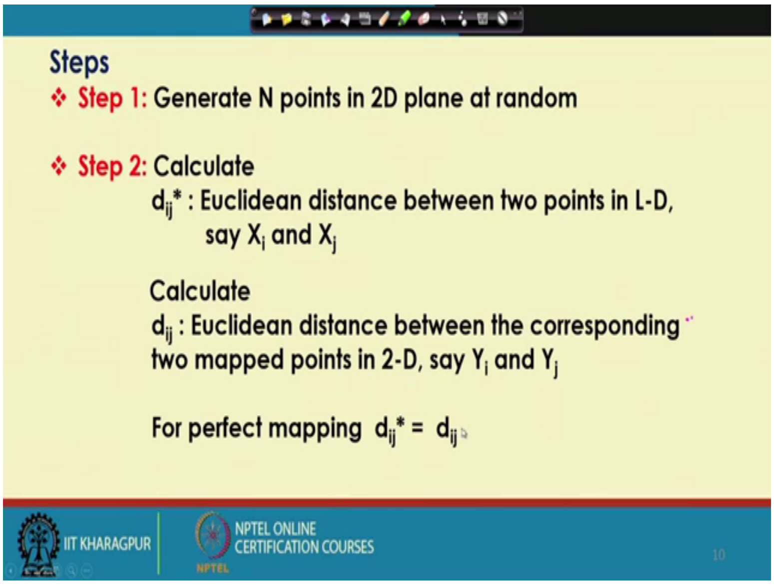
(Refer Slide Time: 07:35)



Now, as I mentioned that in L dimensional space. So, a particular data point if I want to represent. So, I will have to use capital L number of numerical values for example, capital X 1 is a collection of small x 11, small x 12 and so on and the last numerical value is small x 1L. Similarly x 2 can be represented as x 21, small x 22, small x 2l and the last one that is capital X n is represented as small x N1, small x N2 and the last term is small x NL.

Now, this is how to represent capital N number of data points in L D space. Now, similarly in 2D plane, we can represent capital N number of data points as follows for example, the first point is y 1 that is nothing, but y 11, y 12, similarly y 2 is small y 21, small y 22 and the last point that is capital Y n is small y N1, small y N2.

(Refer Slide Time: 09:02)

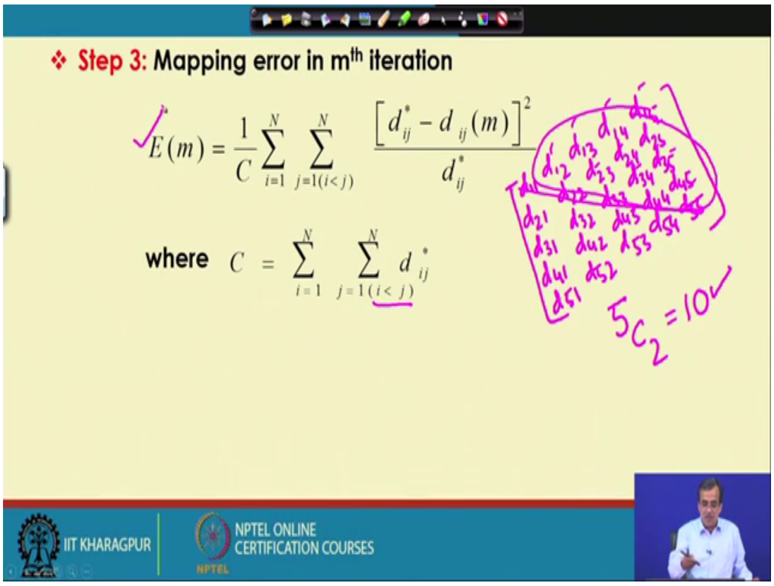


Now, let us see the steps. Now, the problem is we will have to do the mapping from higher dimension to lower dimension. So, step one. So, the information of the higher dimension is known to us of all the data points. Now, what we do is we are going to find out the corresponding data point on the 2D plane.

So, what we do is we generate N data points in 2D plane at random using the random number generator and what you do is we generally a considered a particular range for the random numbers that is 0 to 1. Then step 2 we calculate the Euclidean distance values between 2 points in higher dimension that is L D space. Say the points are X i and x j and their Euclidean distance is nothing, but d ij star. So, d ij star is the Euclidean distance between the 2 data points capital X i and capital X j in higher dimension and this particular information is known to us.

Now, what we do is we generate the data points say i and j in 2D using the random number generator and what you do is we try to calculate the Euclidean distance between the 2 points i and j in 2D plane and that is denoted by d ij. So, we have calculated the Euclidean distance values between 2 points in higher dimension which is known and we have also calculated the Euclidean distance between the 2 corresponding points i and j in lower dimension. Now, for perfect mapping, this d ij star will equal to d ij, but as the data points in 2D are generated at random. So, there is no guarantee that will be getting the perfect mapping. So, there will be some error. Now, this error in mapping will have to minimize using one optimization tool and generally we use actually a steepest descent method.

(Refer Slide Time: 11:36)



Now, I am just going to discuss how to use a steepest descent method to minimize. So, this particular error, but before that let me explain how to calculate this particular the mapping error.

Now, mapping error in mth iteration is nothing, but E m that is 1 by C summation i equals to 1 to N, N is the total number of data points. Summation j equals to 1 to N provided i is less than j. Now, d ij star minus d ij m. Now, this particular d ij star minus d ij m it could be either positive or negative and that is why we put the square sign here just to make it positive divided by d ij star that is the known Euclidean distance values in the higher dimension. So, this is the way actually Sammon represented the error in mapping at mth iteration.

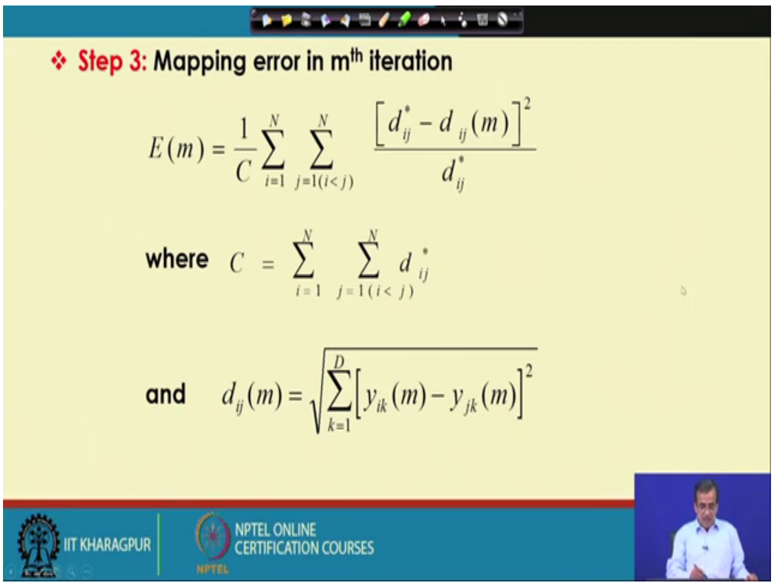
Now here, how to determine this particular C. Now, C is nothing, but a summation i equals to 1 to capital N, summation j equals to 1 to capital N I less than j d ij star. Now, to explain this let me take one numerical example. Now, supposing that I have got 5 data points. Now, if I am got to 25 data points there Euclidean distance matrix can be written as follows. Now, this is d 11, d 12, d 13, d 14, d 15, then d 21, d 22, d 23, d 24, d 25 then comes d 31, d 32, 33, 34, 35 then comes d 41, 42, 43, 44, 45, then comes d 51, d 52, d 53, 54, 55. So, if there are 5 data points. So, we have got 5 multiplied by 5 25 distance values. Now, out of these 25 distance values, this d 11, d 12, d 13, d 14 and d 55 their numerical values are equal to 0. So, this particular diagonal of the matrix will contain all 0 values.

Now, if I concentrate on the right hand side of this particular diagonal. So, we have got 1 2 3 4 5 6 7 8 9 10. So, there are 10 distance values. Now, this particular 10 is nothing, but 5 C 2 and that is nothing, but 10. So, what I do is we consider the Euclidean distance values on one side of this particular the diagonal and d 12 is equal to d 21 similarly d 13 is equals to d 31. So, we did not consider the both the sides of this particular the principle diagonal.

Now that means, if we can calculate only the 10 Euclidean distance values so my purpose will sort, the same thing has been followed here. So, C is nothing, but summation i equals to 1 to N, summation j equals to 1 to N with the condition i is less than j. The moment I put i is less than j. So, I am just going to consider only one side of this particular the diagonal principle diagonal of this matrix and we can find out what should be the distance values the sum of this particular the distance values. The reason we consider the sum of the distance values actually we try to find out the average; that means, this particular thing is multiplied by 1 by C. So, we will be getting the expression for this error in mapping at mth iteration.

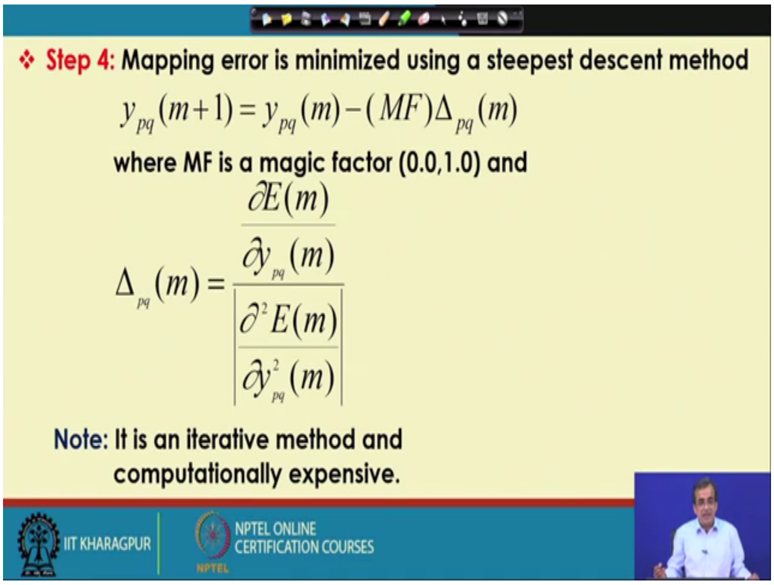
Now, let us see with this how to proceed further.

(Refer Slide Time: 16:54)



And here this d ij m, this particular d ij m is nothing, but the Euclidean distance between the 2 data points i and j corresponding to mth iteration in lower dimension and that is nothing, but square root summation K equals to 1 to d y i K m minus y j K m square of that. So, this particular Euclidean distance d ij m that is in lower dimension and as I mentioned d ij star the Euclidean distance between the 2 data points in higher dimension. So, this is the way we express this particular error in mapping and what you do is in step 4 this mapping error is minimized using a steepest descent method because this particular method on principal cancel the minimization problem and for this steepest descent method. So, this is the rule for updating.

(Refer Slide Time: 17:42)



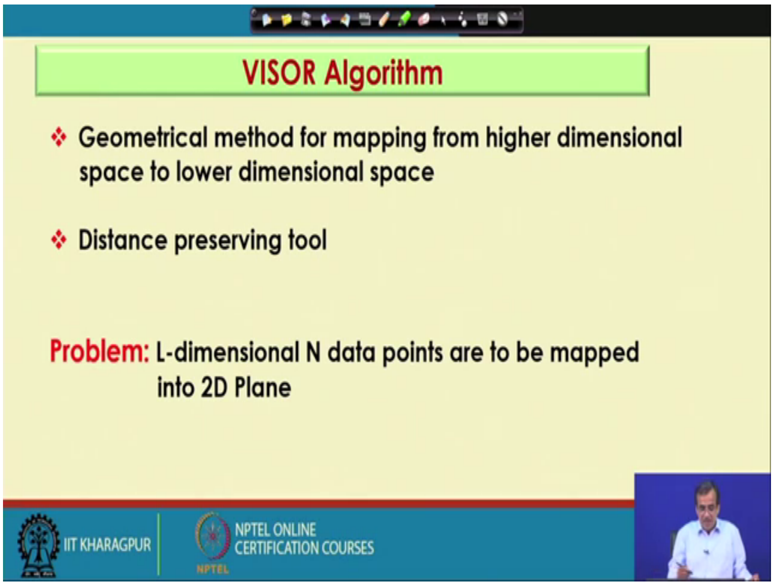
Like y pq m plus 1 is nothing, but y pq m minus MF is the magic factor multiplied by delta pq m. Now, here this magic factor is nothing, but the step length. If you remember that in steepest descent method we use a steep a step length and the search direction. So, here the step length is denoted by this magic factor MF and it varies between 0 and 1, and this particular delta pq m is nothing, but the search direction that is a gradient direction and we put one negative sign because we move in a direction opposite to the gradient because this is a minimization problem.

Now, is delta pq m that is actually the search direction that is a gradient is nothing, but the partial derivative of E m with respect to y pq m divided by we consider the norm of the second order differentiation. And this particular norm will provide some numerical value and ultimately this particular delta pq m is going to indicate the search the gradient direction and the search will be opposite to the gradient. So, using this particular rule for updating this steepest descent method can minimize the error in mapping and this particular method is an iterative method initially there will be a large amount of error because the data points in lower dimension are generated at random so we have got no control on this particular data point and it is expected that initially there will be a large amount of error and through a large number of iteration. So, this steepest descent algorithm is going to minimize so that particular the error in mapping.

Now, our experience says that if we run this particular algorithm say about 100 iteration there is a possibility that will be getting very accurate error corresponding to that the error in mapping will be very less.

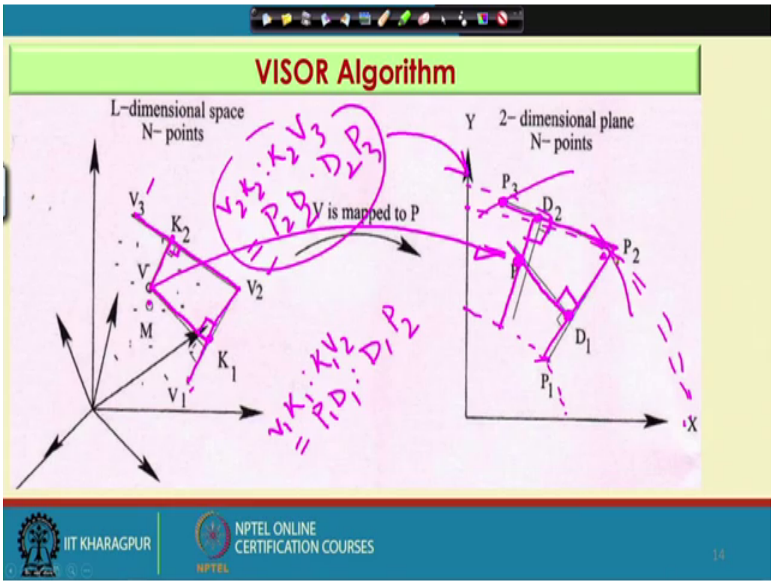
Now, as I told that this is actually a mathematical approach and we can minimize this particular the error in mapping. Now, I am just going to start with another algorithm which is a grab base techniques and this particular technique is known as the VISOR algorithm.

(Refer Slide Time: 21:02)



Now, here actually what I do is we try to use the geometrical information of this particular the higher dimensional space and the data are initially there in the higher dimension and we want to do the mapping to the lower dimension. And once again this is a distance preserving tool not a topology preserving technique. The same problem we are going to tackle like we have got capital N number of data points and this data points actually we will have to do the mapping to the lower dimension; that means, from higher dimension to lower dimension, so capital N number of data points, we will have to do the mapping.

(Refer Slide Time: 21:41)



Now, how to do it, so I am just going to discuss in details. Now, supposing that we have got the problem like this we have got capital N number of data points in a higher dimension that is L D space. Now, this is how to represent the L D space.

Now, we have got say capital N number of data points and as I mention to represent a particular data points. So, I need capital L number of numerical values. Now, let us see how to do this particular the mapping. Now, here I am just going to do the mapping from L dimension to the 2 dimension; that means, I will have to find out capital N number of data points on the 2D plane. Let us see how to do it.

Now, the steps I am just going to discuss first in details. Now, what will have to do is, I have got capital N number of data points the first thing we do is we try to find out the centroid of this particular capital N data points. Now, centroid is denoted by. So, this particular M. How to find out the centroid? Now, for a particular data point we have got capital L number of numerical values we concentrate on the first dimension. So, he will be getting for N data points. So, N number of the numerical values corresponding to the first dimension. So, we add all the capital N numerical values and then we divided by the capital N that will be the first dimension information for this particular the centroid point. The same procedure we repeat for the second dimension, third dimension up to the Lth dimension and we can find out the coordinate of this particular the centroid point that is M.

Now, once we have got the centroid point. Now, what we do is we try to locate 3 pivot points like V 1, V 2 and V 3. Now, I am just going to discuss how to locate this particular the V pivot points. Now, this pivot points are actually going to help while doing this particular the mapping. Now, to look at the pivot points say V 1 what I do is. So, starting from the centroid we try to find out the Euclidean distance of all capital N data points. So, there will be capital N number of numerical values for the Euclidean distance. We compare those numerical values of the Euclidean distance and we try to find out a data point which is for this from this particular centroid point; that means, the Euclidean distance should be the maximum.

Supposing that V 1 is the data point whose Euclidean distance is maximum from M. So, V 1 is selected as the first pivot point and once we have got this particular first pivot point. Now, I have got remaining N minus 1 data points. So, what you do is we calculate the Euclidean distance values starting from V 1 to all N minus 1 data points, so there will be n minus one Euclidean distance values and one second we compare those n minus one Euclidean distance values and we try to locate a point that is a second pivot vector for which we get the maximum Euclidean distance values starting from V 1.

So, I will be getting this particular the V 2 and once I have got this particular V 2, now we try to find out another pivot point. Now, this third pivot point should be should have the maximum distance both from V 1 as well as V 2. So, what you do is out of capital N data points 2 points we have already selected. Now, we have got remaining capital N minus 2 data points. So, what you do is for this remaining capital N minus 2 data points, so I calculate the Euclidean distance from V 1 and Euclidean distance from V 2 and we try to locate a point out of this capital N minus 2 data points whose Euclidean distance will be maximum both from V 1 as well as V 2 and we try to locate this particular the third pivot point that is V 3.

Now, till now, whatever we have got is 3 pivot points V 1, V 2 and V 3 and once I have got this particular the 3 pivot points. So, we try to locate the corresponding 3 pivot points in 2 dimension. So, what you do is, we try to find out the Euclidean distance from the origin of this higher dimension to V 1 Euclidean distance from origin to V 2 Euclidean distance from the origin to V 3. So, we will be getting 3 numerical values and considering those 3 numerical values as the radius. So, we draw the circular arc on the 2D plane. Now, the first circular arc could be approximately something like this, the second one it could be something like this, the third one could be something like this. So, we can find out 3 such the circular arcs.

Now, once I have got this particular 3 circular arcs. So, what you can do is I can also calculate the Euclidean distance value between V 1 V 2 and that between V 2 and V 3. So, on the first circular arc any point we select at random supposing that I am selecting this particular point. Now, I know the Euclidean distance between V 1 and V 2 keeping that. So, what I do is and considering P 1 as the center. So, we draw one circular arc on the second this circular arc and I will be getting a data point here that is nothing, but P 2 and once I have got this particular P 2. Now, I can find out the Euclidean distance between V 2 and V 3.

So, we consider P 2 as center and the Euclidean distance between V 2 and V 3 as the radius we draw one circular arc and you will be getting this particular point P 3. So, corresponding to V 1, V 2 and V 3 in higher dimension I am getting P 1, P 2 and P 3 these 3 pivot vectors in lower dimension. Now, once we have got this, now how to proceed.

So, what I do is in a higher dimension we join this point V 1 and V 2 by a straight line like this similarly V 2 and V 3 a joint by another straight line something like this like this. The same thing we do in lower dimension. So, we join the line between P 1 and P 2 like this and join the line between P 2 and P 3 like this. Now, what you do is say we want to do the mapping for a particular point say V from higher dimension to lower dimension; that means, I am just going to find out a corresponding point in 2D corresponding to this particular V in higher dimension a corresponding point in 2D. So, what I do is from this particular V we draw the perpendicular on V 1 V 2. So, we draw the perpendicular something like this. So, this is perpendicular.

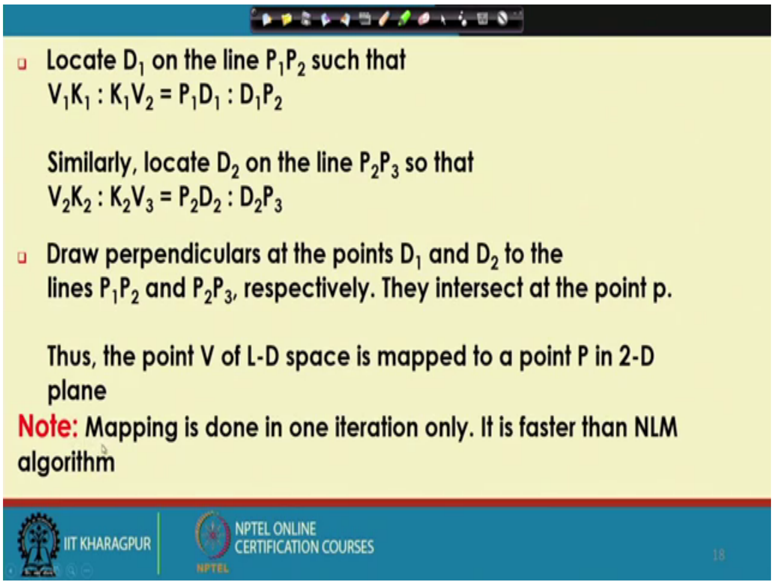
Similarly, we draw the perpendicular on V 2 V 3 like this. So, this is perpendicular and we try to locate the points like K 1 and K 2 and once I have got this particular K 1 I can write down I can find out V 1 K 1 ratio K 1 V 2, K 1 V 2 and this we keep the same here like P 1 D 1, P 1 D 1 ratio like your D 1 P 2 and following this particular relationship we try to locate the point D 1 on P 1 P 2.

The same principle we follow here also. So, we have got this particular point K 2. Now, we consider the ratio V 2 K 2 ratio this K 2 V 3 is nothing, but P 2 D 2, P 2 D 2 ratio D 2 P 3. Now, this corresponds. So, this particular this particular point d 2. Now, we will be able to locate this particular point D 2 on the line P 2 P 3. Now, once we have got this particular D 1 and D 2. So, what we do is we draw the perpendicular at the point D 1 to the line P 1 P 2. So, I will be getting this type of perpendicular line. Similarly at the point D 2 we draw another perpendicular to the line P 2 P 3 and will be getting another perpendicular and these 2 perpendicular lines are going to intersect at a particular point P. So, this P is actually nothing, but the point V in higher dimension.

So, this particular point V in L D space is map to the point P into 2D plane and the same principle. So, we will have to follow for the remaining data points. Now, already we have got say 1 2 3 4 data points in lower dimension. So, out of this capital N the 4 data points we have already mapped to the lower dimension the remaining N minus 4 data points are to be mapped following the principal the way I discuss how to map the point V to point P. So, we follow this particular principal for the remaining data points so that all the capital L data points in L dimensional space can be mapped to N data points on 2D plane. So, this is the way actually we do the mapping approximately using the principle of this particular the visor algorithm.

Now, this particular algorithm are the steps I have already written it here, but all such steps are already discussed in details. So, whatever I discussed, all such things I have written it here.

(Refer Slide Time: 34:18)



Now, here we just put one note that this here the mapping is done in one iteration only and this is very fast. It is in fact, much faster compared to the Sammon's non-linear mapping. Now, what will be the quality of the map data points that we will be discussing after sometime.

Thank you.