Computer Vision and Image Processing - Fundamentals and Applications

Professor. Dr. M.K. Bhuyan
Department of Electronics and Electrical Engineering
Indian Institute of Technology, Guwahati

Lecture No. 34
Introduction to Machine Learning - V

Welcome to NPTEL MOOC's course on Computer vision and Image Processing- Fundamental and Applications. In my image transformation class, I discussed the concept of PCA, the Principle Component Analysis. So, how to reduce the dimension of the input vector. If I consider the feature vector, suppose X; I can reduce the dimension of the feature vector, by neglecting the redundant information by PCA; the Principal Component Analysis. PCA finds the greatest variance of data.

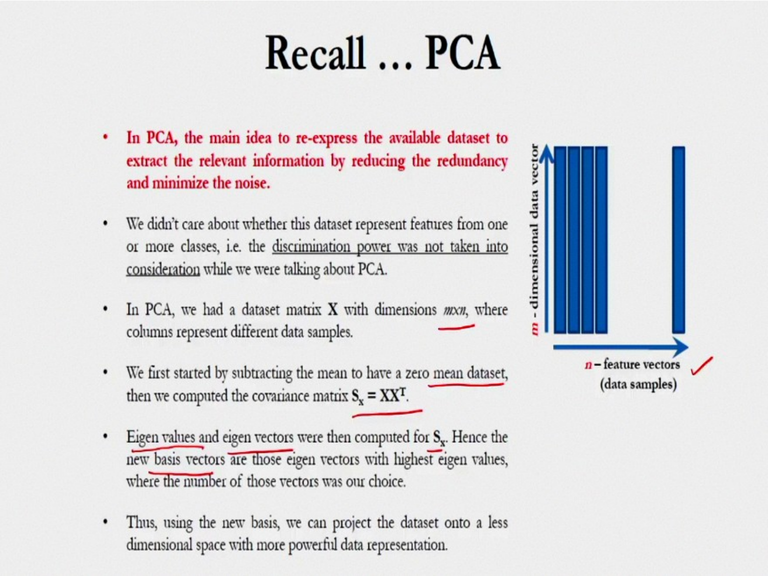
But one problem with the PCA is that, it does not consider the class information. Suppose I have number of classes, and the discrimination between the classes, that information is not considered by the PCA. So, for this, I will consider another method, that is called the Linear Discriminate Analysis. So, in case of Linear Discriminate Analysis, I can reduce the dimension of the input vector. Also, I can find the separation between the classes; that is the discrimination between the classes, I can do. So, that concept I am going to discuss today.

And also, I discuss the concept of the Bayesian decision making; the Bayesian classifier. That is nothing but the generative model. So, what is the concept of generative model? That means, I have the information of the class conditional density. The probability of X given w, z. So, that information is available, and with that information I can do the classification. So, I can determine the posterior density; the density is probability of w, z given X; I can determine.

There is another classifier that is called discriminative classifier. So, in this case, the information of the class conditional density is not important. So, I can find the best decision boundary between the classes. Suppose, if I consider 2 classes, I can find the best decision boundary between these 2 classes. So, for this I will discuss one algorithm that is the Support Vector Machine.

So, today's class I will discuss these 2 concepts; one is the Linear Discriminant Analysis, and another one is the Support Vector Machine. So first, let us consider the concept of the LDA. And, what are the problems with the PCA? That concept I am going to explain.

(Refer Slide Time: 03:00)



Recall … PCA

- **In PCA, the main idea to re-express the available dataset to extract the relevant information by reducing the redundancy and minimize the noise.**
- We didn't care about whether this dataset represent features from one or more classes, i.e. the discrimination power was not taken into consideration while we were talking about PCA.
- In PCA, we had a dataset matrix **X** with dimensions *mxn*, where columns represent different data samples.
- We first started by subtracting the mean to have a zero mean dataset, then we computed the covariance matrix $S_x = XX^T$.
- Eigen values and eigen vectors were then computed for S_x. Hence the new basis vectors are those eigen vectors with highest eigen values, where the number of those vectors was our choice.
- Thus, using the new basis, we can project the dataset onto a less dimensional space with more powerful data representation.

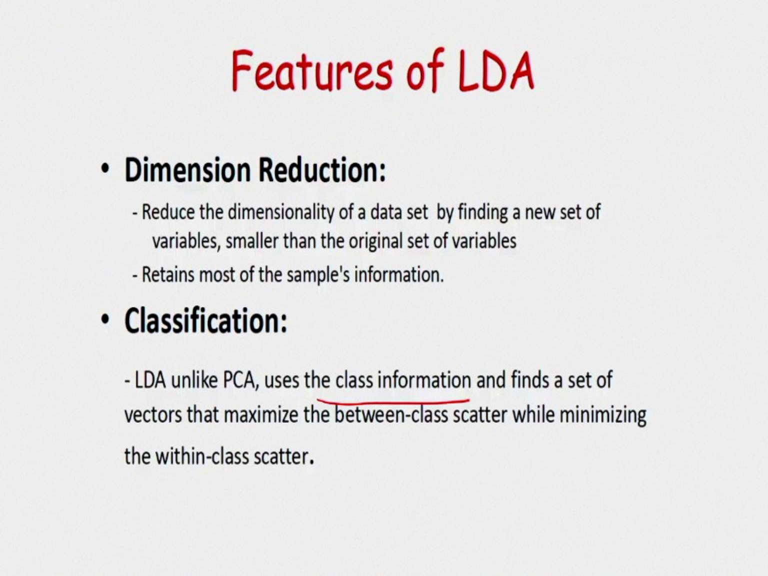
So, in case of the PCA, if you know, that I discuss in the image transformation class; I can reduce the dimension of the feature vector or the input vector. But in this case, I am not considering that discrimination between the classes, that information is not available in the PCA. So, in this figure, you can see I am considering n number of feature vectors. And, I am considering the m-dimensional vector, that I am considering. So that means, the dataset matrix X has a dimension of m × n.

And for this PCA, the method is like this. First, I have to subtract the mean from the original data. So that, I will be getting a zero-mean dataset. So, I will be getting a zero-mean dataset. And after this, I have to compute the covariance matrix. The covariance is matrix, I can compute like this. And, from the covariance matrix, I can determine the transformation matrix. So, for this, I have to determine the eigenvectors, and also I have the corresponding eigenvalues. So, this transformation matrix for the principal component analysis, I can determine from the eigenvectors of the covariance matrix.

So, you can see, I can determine the eigenvalues, and the eigenvectors I can determine from the covariance matrix. And from this, I can determine the transformation matrix. That is the basis vector, I will be getting. And, I can consider the highest eigenvalue, I can consider; and the corresponding eigenvectors, I can consider. So, that concept already I have explained in my PCA class, that is in the image transformation.

But one problem of the PCA, that already I have highlighted, that is the class discrimination information is not available. That is only I can reduce the dimension of the input vector, the input data, or the input feature vector.

(Refer Slide Time: 05:00)



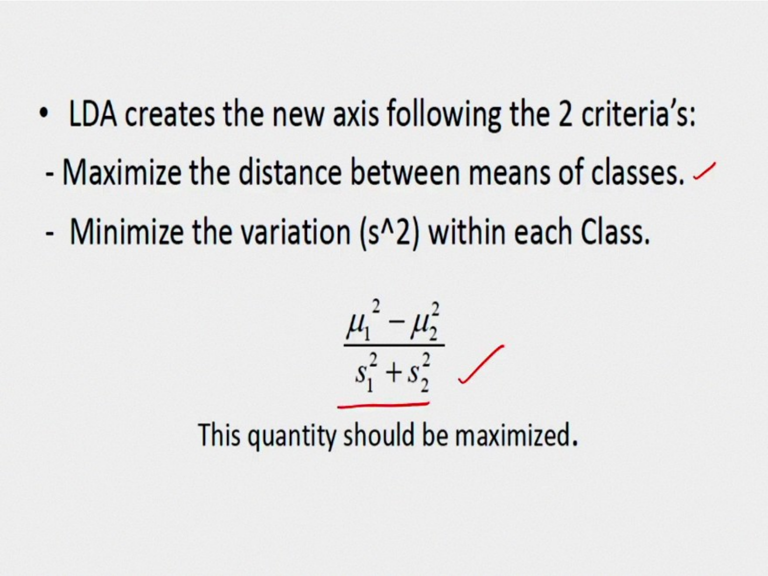
Features of LDA

- **Dimension Reduction:**
  - Reduce the dimensionality of a data set by finding a new set of variables, smaller than the original set of variables
  - Retains most of the sample's information.
- **Classification:**
  - LDA unlike PCA, uses the class information and finds a set of vectors that maximize the between-class scatter while minimizing the within-class scatter.

So in case, of the LDA, the Linear Discriminant analysis, I can reduce the dimension of the data, the input data. So, reduce the dimensionality of a data set, by finding a new set of variables, smaller than the original set of variables. So, I can do this. And also, I can retain most of the sample's information. So, redundant information I can neglect, but I can retain most of the sample's information.

So, unlike PCA, LDA uses the class information. So, that information is available in case of the Linear Discriminant Analysis. And, I have to find a set of vectors, that maximize the between-class scatter, while minimizing the within-class scatter matrix. So, that concept I am going to explain. Because, in case of the Linear Discriminant Analysis, I have the class information, and I have to find a set of vectors, that maximize the between-class scatter, and I can also minimize the within-class scatter. So, this mathematical concept I am going to explain in case of the LDA, the Linear Discriminant Analysis.

(Refer Slide Time: 06:15)



- LDA creates the new axis following the 2 criteria's:

- Maximize the distance between means of classes. ✓
- Minimize the variation (s^2) within each Class.

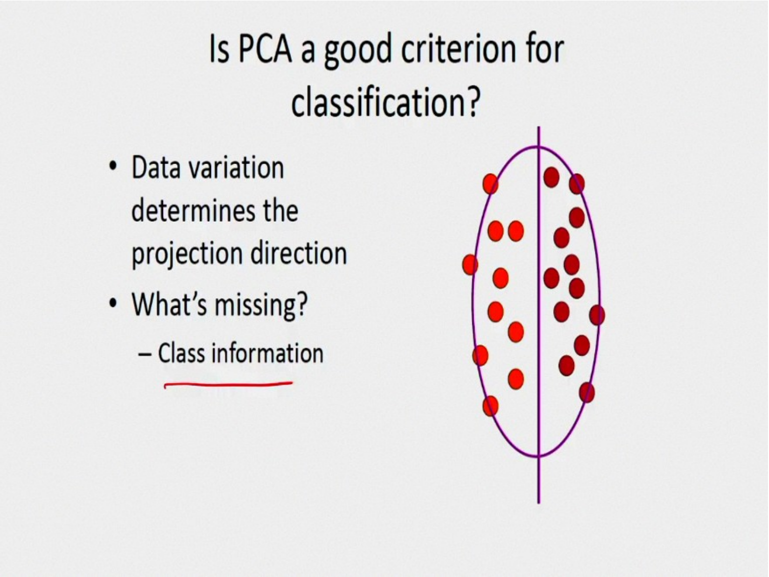
$$\frac{\mu_1^2 - \mu_2^2}{\underline{s_1^2 + s_2^2}} \checkmark$$

This quantity should be maximized.

So, LDA finds the new axis based on these 2 criteria. So, one is the maximize the distance between the means of the classes. So, I can consider, suppose 2 classes. So, I can maximize the distance between these classes. So, that means, I can find the maximum distance between the means of the classes, and also the minimize the variation within the class; that I can consider. So, one is the maximize the distance between means of the classes, and also, I can minimize the variation within the class.

So, I can consider this one, that means I have to minimize the variation within the class, and also, I have to maximize the distance between that means of the classes. So, this quantity I have to maximize.

(Refer Slide Time: 07:07)

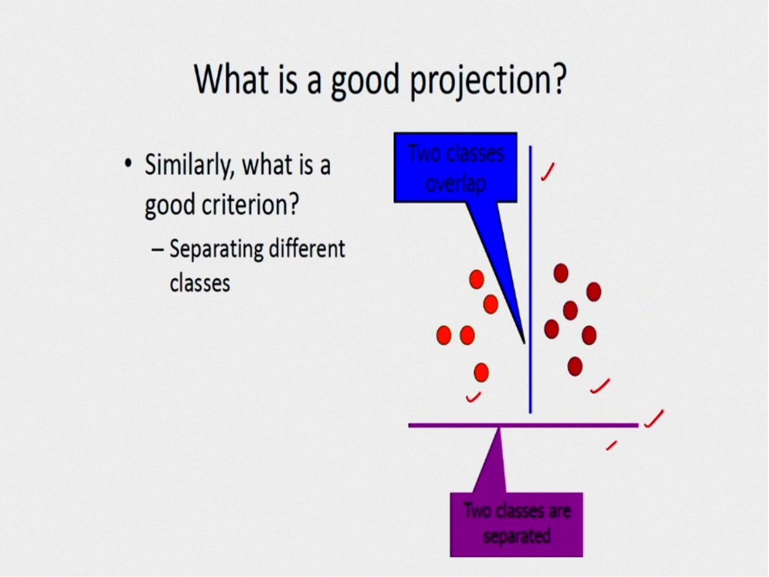


Is PCA a good criterion for classification?

- Data variation determines the projection direction
- What's missing?
  - Class information

Now, the question is; is PCA a good criterion for classification? Now in case of the PCA, the PCA finds direction of greatest variance. Data variation determines the projection direction, but in case of the PCA, the class information is missing. We do not have the class information, but how actually we consider. We consider the eigenvectors, that means we want to find the directions of greatest variation, that means we can find the eigenvectors of the covariance matrix. But the class information is missing in case of the PCA.

(Refer Slide Time: 07:46)



What is a good projection?

- Similarly, what is a good criterion?
  - Separating different classes

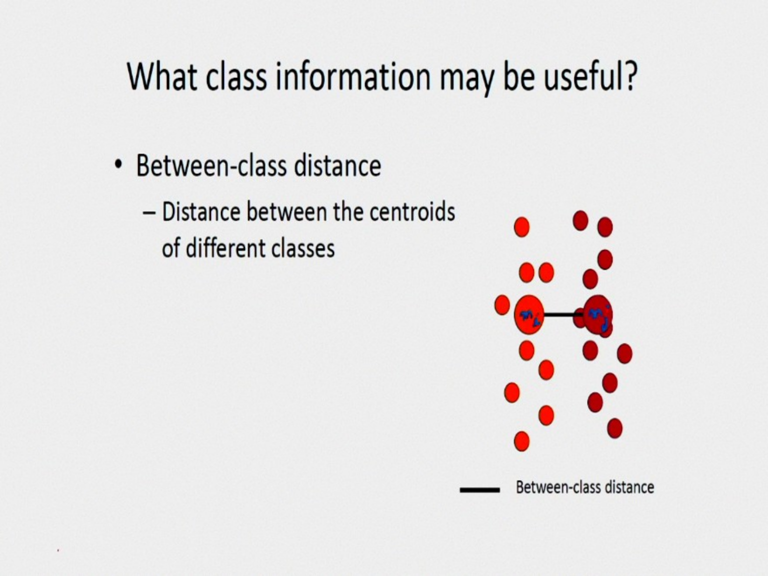
Two classes overlap

Two classes are separated

So, let us consider, what is the projection? In case of the PCA, we consider the eigenvectors, that is the direction of the projection. Now let us consider, what is the good projection. Here, in this figure, you can see I am considering 2 classes. You can see, one is the, this class, another one is the, this class, the 2 classes. And I want to find the good projection. In case of the blue projection, if I consider the blue projection line, there may be overlapping of the samples of different classes. But, if I consider this projection, that means, these 2 classes will be well separated.

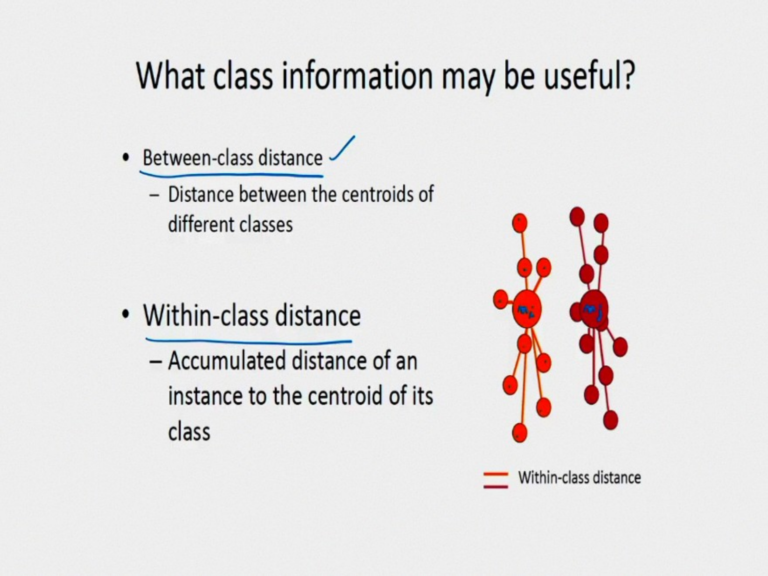
In the first projection, if I consider the first projection, that is the, the blue projection, then in this case, these 2 classes are not well separated. But, in the second case, if I consider, the second projection, if I consider this projection, the 2 classes are separated. So, that means in case of the LDA, I have to see this condition, that is the separation between the classes.

(Refer Slide Time: 08:48)



What class information may be useful?

- Between-class distance
  - Distance between the centroids of different classes



What class information may be useful?

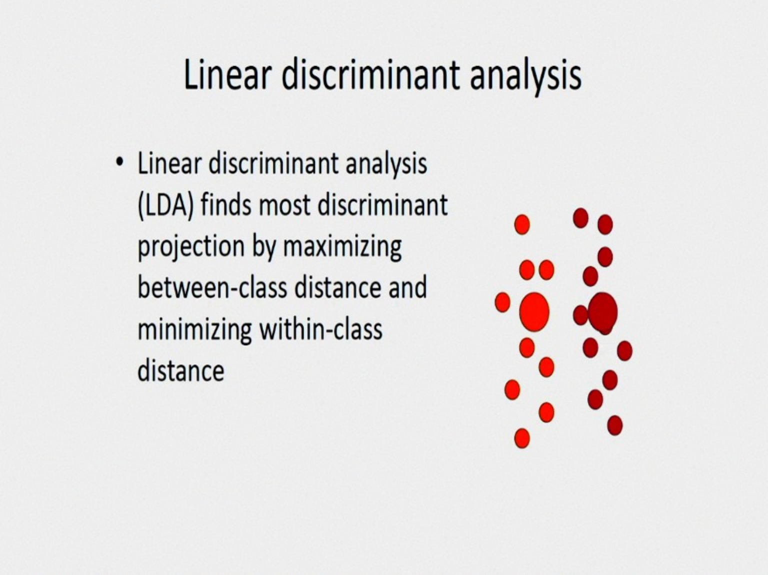
- Between-class distance
  - Distance between the centroids of different classes
- Within-class distance
  - Accumulated distance of an instance to the centroid of its class

So, for this, this information is important; one is the between-class distance. So here, again I am showing 2 classes, you can see. So, this is the centroid of suppose class i, and this is the centroid of another class m_j; that is the mean. Now, in this case the between-class distance should be maximum. So, you can see between these 2 means; one is m_i, another one is m_j. So, one is m_i, and another one is m_j. These 2 means I am considering. For 2 classes, the distance between the centroid of different classes should be maximum. That is between-class distance.

After this, I am considering another information, and that is within-class distance. So that means, it is the accumulated distance of an instance to the centroid of its class. So, that means, if I consider, this is a centroid, the centroid is $m_i$and m_j. And you can see, I am considering the sample points, corresponding to the centroid m_i. So, this within-class distance should be minimum.

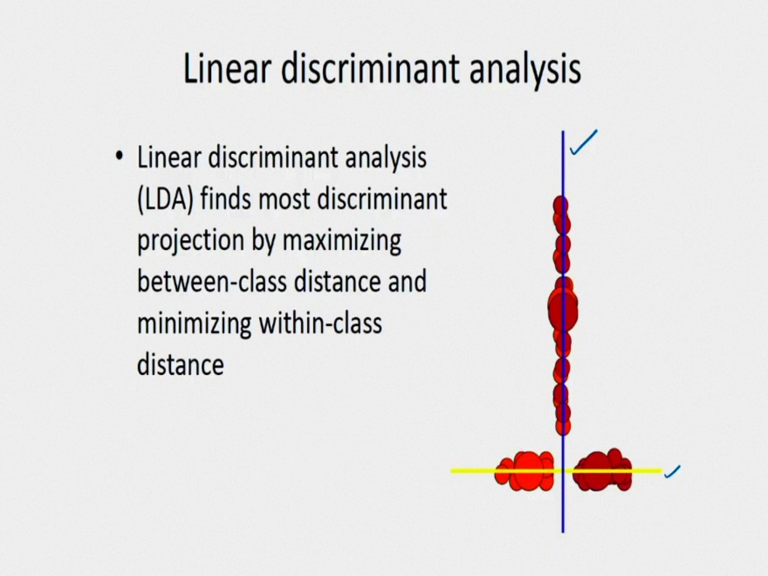
So, suppose distance between these samples and the centroid, I can determine, and I can determine the accumulated distance. So, that should be minimum, that corresponds to within-class distance. So, that means for the LDA, this is important; one is the between-class distance that is important, distance between the centroid of different classes that should be maximum. And within-class distance, that means accumulated distance of an instance to the centroid of its class. So, that should be a minimum. So, these 2 conditions, one is the within-class distance, another one is the between-class distance corresponding to LDA and that is very important.

(Refer Slide Time: 10:52)



Linear discriminant analysis

- Linear discriminant analysis (LDA) finds most discriminant projection by maximizing between-class distance and minimizing within-class distance



Linear discriminant analysis

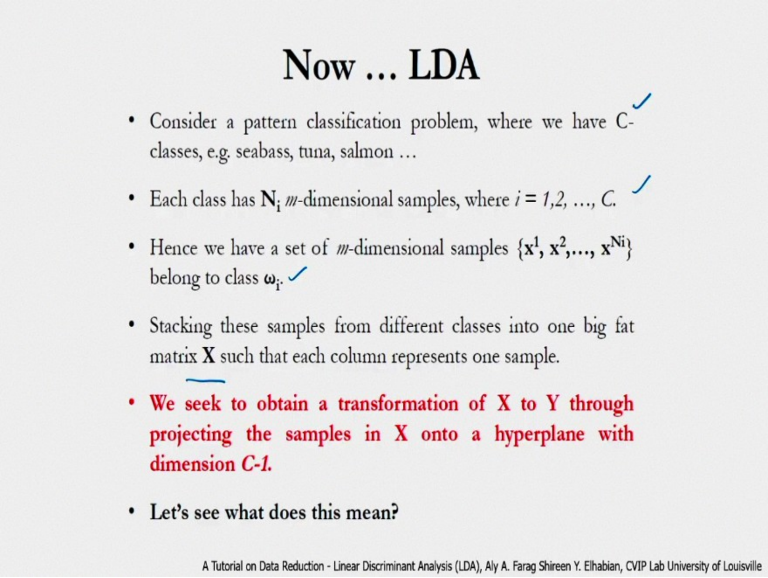
- Linear discriminant analysis (LDA) finds most discriminant projection by maximizing between-class distance and minimizing within-class distance

So, in case of the Linear Discriminant Analysis; LDA finds most discriminant projection by maximizing between-class distance and minimizing within-class distance. So, here I am showing these 2 cases. First you can see, I am considering; this is the projection direction, that is the blue is the projection direction. And you can see, the classes, the samples of the classes, 2 classes are overlapping, that is the discrimination between these 2 classes is minimum. But, if I consider the second projection direction, that is the yellow, the discrimination between these 2 classes is maximum.

So, you can see, the discrimination between these 2 classes is maximum. So, I have to find that direction, in which direction, the discrimination between these classes will be maximum. So, that direction I have to estimate. So, for this I have to consider these 2 cases; one is the within-class distance, another one is the between-class distance, I have to consider. So that means, I have to maximize between-class distance, and I have to minimize within-class distance.

So, based on this, I have to find the projection direction. And based on this projection direction, I can find maximum separability between these 2 classes. Now, I am considering 2 classes. It may be applicable for more than 2 classes also. So, if I consider C number of classes. So this concept is also applicable. But in this example, I am only considering 2 classes.

(Refer Slide Time: 12:26)



Now … LDA

- Consider a pattern classification problem, where we have C-classes, e.g. seabass, tuna, salmon … ✓
- Each class has $\mathbf{N}_i$ m-dimensional samples, where $i = 1, 2, \ldots, C$. ✓
- Hence we have a set of m-dimensional samples $\{\mathbf{x}^1, \mathbf{x}^2, \ldots, \mathbf{x}^{N_i}\}$ belong to class $\boldsymbol{\omega}_i$. ✓
- Stacking these samples from different classes into one big fat matrix **X** such that each column represents one sample.
- **We seek to obtain a transformation of X to Y through projecting the samples in X onto a hyperplane with dimension *C-1*.**
- **Let's see what does this mean?**

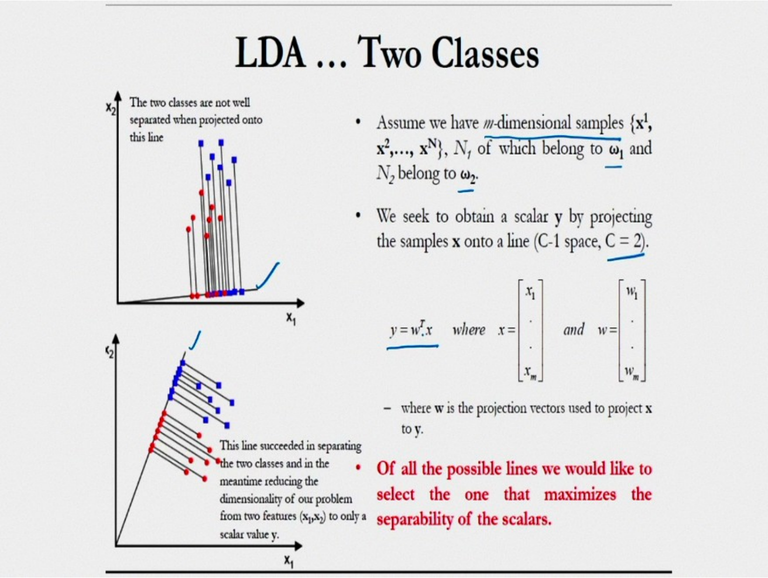
A Tutorial on Data Reduction - Linear Discriminant Analysis (LDA), Aly A. Farag Shireen Y. Elhabian, CVIP Lab University of Louisville

So, what is the mathematics behind LDA, that I want to explain. So, let us consider a pattern classification problem, and for this, I am considering C number of classes, I am considering. So, maybe the classes, maybe the fishes; like the, seabass, tuna, salmon, like this I can consider number of classes, the C number of classes I can consider. And each class has N_i samples. So, m-dimensional samples are available. And how many samples are available? N_i number of samples for each of the classes. And we have a set of m-dimensional samples.

So, we have a set of m-dimensional samples. So, corresponding to the class, the class is w_i. I have the samples x_1, x_2 like this. So, we have Ni number of samples, and it is the m-dimensional samples. And from this, I can get a matrix, the matrix is X. That is stacking these samples from different classes into 1 matrix, that matrix is the X. And this is column of matrix represent 1 sample. So, I will be getting a matrix, the matrix is X, from all the samples of different classes.

Now, I want to find a transformation of X to Y; X is a input data vector, suppose. So, I want to find a transformation of X to Y through projecting the samples in X onto a hyperplane with the dimension C minus 1. So, I have to find the projection direction, that new data will be Y, after the projection, and the objective is to get the maximum discrimination between the classes. So, I have to find the best projection direction, I have to find.

(Refer Slide Time: 14:27)



LDA … Two Classes

The two classes are not well separated when projected onto this line

This line succeeded in separating the two classes and in the meantime reducing the dimensionality of our problem from two features (x_1,x_2) to only a scalar value y.

- Assume we have *m*-dimensional samples $\{\mathbf{x}^1, \mathbf{x}^2,\ldots, \mathbf{x}^N\}$, N_1 of which belong to ω_1 and N_2 belong to ω_2.
- We seek to obtain a scalar **y** by projecting the samples **x** onto a line (C-1 space, C = 2).

$$y = w^T x \quad where \quad x = \begin{bmatrix} x_1 \\ . \\ . \\ x_m \end{bmatrix} \quad and \quad w = \begin{bmatrix} w_1 \\ . \\ . \\ w_m \end{bmatrix}$$

- where **w** is the projection vectors used to project **x** to y.

- **Of all the possible lines we would like to select the one that maximizes the separability of the scalars.**

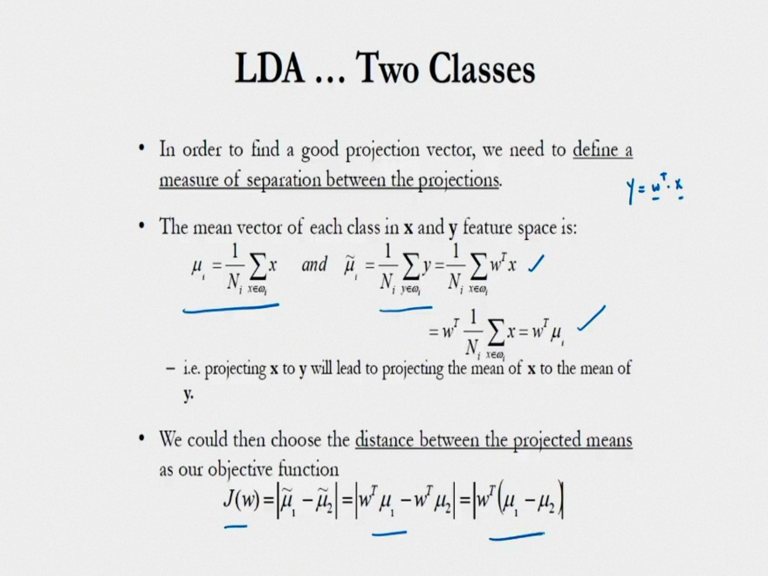
For simplicity, I am now considering only 2 classes suppose. So, this principle can be extended for C number of classes. So first, I am considering 2 classes. So, we have the m-dimensional samples. So, we have the m-dimensional samples, and we have the N number of samples. So, the N_1 number of samples belong to the first class; the first class is w_1. And N_2 number of samples belonging to another class, another class is w_2.

And, we seek to obtain a scalar y by projecting the samples x onto a line. So, in this case, we have we are considering 2 number of classes, that means C is equal to 2. So that means, C minus 1 means, it is 2 minus 1; it is 1. So, dimension is reduced to 1. And what I am getting? I am just doing the projection w ^{T} x; that is actually the dot product, if I consider the vector form. So, it will be the dot product. So, y will be the scalar. So, w ^{T}. x. So, w is the projection vector, and x is the input vector, that I am considering. So, if I take the dot product between w ^{T} and x. So, I will be getting the scalar, the scalar is y.

In the figure, you can see, I am showing a projection direction, the direction is this, one projection direction you can see. And in this case, also I am considering 2 classes, and here I am considering 2-dimensional samples. Because I am considering x1 and x2. This is 2-dimensional samples. So, corresponding to this you can see the separation between the classes is minimum. Because there is an overlapping between the samples of the classes, the 2 classes.

But, if I consider, in the second figure, I am considering a projection direction; you can see, I am getting the separation between the classes, between the samples of the 2 classes. So, that means the second projection direction is better as compared to the first projection direction. So, I have to find, which one is the best projection direction. So, that is the objective of the LDA.

(Refer Slide Time: 16:40)



LDA ... Two Classes

- In order to find a good projection vector, we need to define a measure of separation between the projections.

$y = w^T . x$

- The mean vector of each class in **x** and **y** feature space is:

$$\mu_i = \frac{1}{N_i}\sum_{x\in\omega_i} x \quad and \quad \tilde{\mu}_i = \frac{1}{N_i}\sum_{y\in\omega_i} y = \frac{1}{N_i}\sum_{x\in\omega_i} w^T x \ ✓$$

$$= w^T \frac{1}{N_i}\sum_{x\in\omega_i} x = w^T \mu_i \ ✓$$

  - i.e. projecting **x** to **y** will lead to projecting the mean of **x** to the mean of **y**.

- We could then choose the distance between the projected means as our objective function

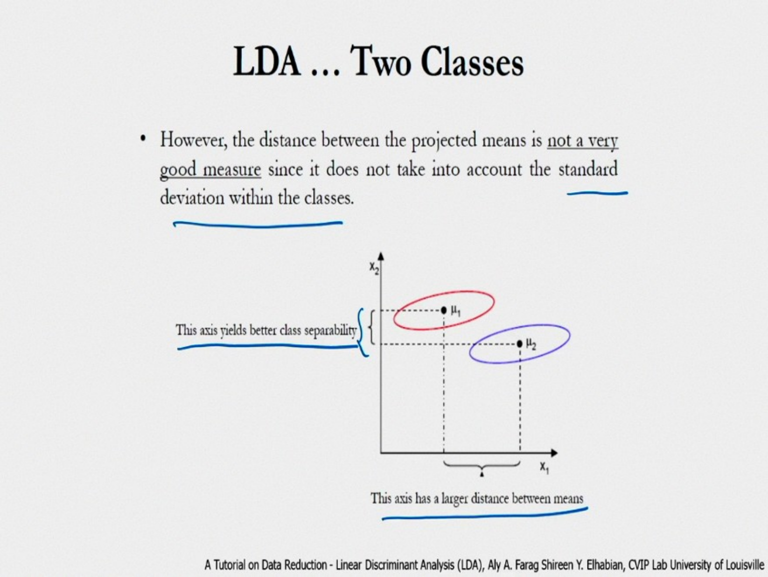
$$J(w) = \left|\tilde{\mu}_1 - \tilde{\mu}_2\right| = \left|w^T \mu_1 - w^T \mu_2\right| = \left|w^T(\mu_1 - \mu_2)\right|$$

So, in order to find a good projection vector, we need to define a measure of separation between the projections. Because, I am getting the projection. The projection is nothing but y. That is nothing but w ^{T} . *x*. So, I am getting the scalar y. So, first the mean vector of each class in x and y feature space, I can determine. So, μ_i, I can determine; because I have N$_i$ number of samples. So, I can determine the mean of x corresponding to a particular class. And you can see I can also determine the mean of the projected data. So, y is the projected data.

So, $\widetilde{\mu}_i$, that I can determine. So, it will be something like this, $\frac{1}{N_i}\sum x^T x$. So, it is, $\widetilde{\mu}_i$, I can also determine. So, you can see it is $\frac{1}{N_i}$ $w^T x$. And after this, just you do this mathematics so I can determine the mean of the projected data. Now, I am considering the objective function. The objective function is J w, and the main objective, or the main goal is to find a maximum distance between the projected means. So that, I will be getting maximum separation between the classes.

So, objective function J w, I am considering, and this is μ_1 is the projected mean for the class 1. And μ_2 is the projected mean, for the class 2. So, I am considering $\widetilde{\mu_1}$and $\widetilde{\mu_2}$, that is the projected mean. So, from this you can determine this. From the previous equation, you can determine this.

(Refer Slide Time: 18:32)



LDA … Two Classes

- However, the distance between the projected means is not a very good measure since it does not take into account the standard deviation within the classes.

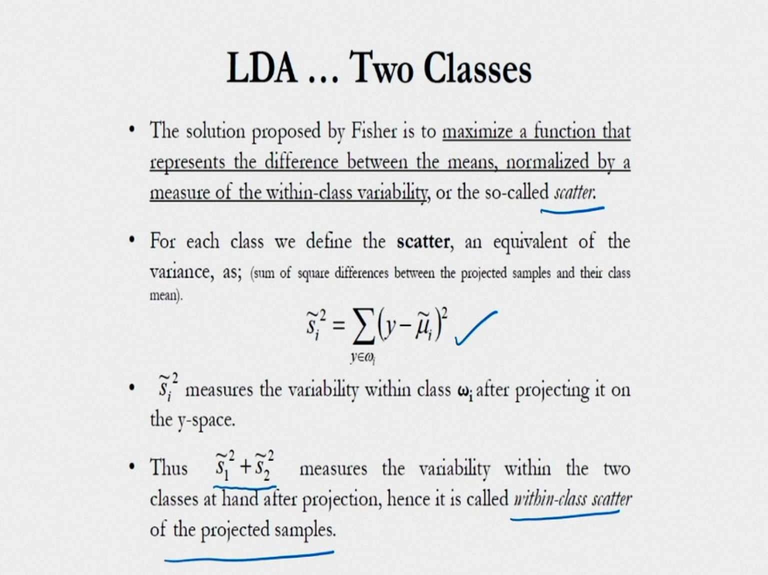
x_2
μ_1
μ_2
x_1
This axis yields better class separability
This axis has a larger distance between means

A Tutorial on Data Reduction - Linear Discriminant Analysis (LDA), Aly A. Farag Shireen Y. Elhabian, CVIP Lab University of Louisville

Now in this case, you can see the distance between the projected mean is not a very good measure. Because, it does not take into account the standard deviation within the class. So, that information is not considered in this case. Because we considered the distance between the projected mean, and that may not be a good measure. Because in this case, we are not considering the standard deviation within the class. So, that information we are not considering.

So, pictorially that concept I am showing here. Here you can see, the axis has a larger distance between the means, in the first case. But in this case, it is not a good separability. There may be some overlapping between the classes. But if I consider the second case; this axis gives better class separability. So, you can see if I consider this axis, that is, this projection direction, then I will be getting maximum separability between the classes. But, in the first case, I will be getting the larger distance between the mean. But in this case, the separability is not good. The separability between the classes is not good as compared to the second case. In the second case, I am getting better class separability.

(Refer Slide Time: 19:48)



LDA … Two Classes

- The solution proposed by Fisher is to maximize a function that represents the difference between the means, normalized by a measure of the within-class variability, or the so-called *scatter.*
- For each class we define the **scatter**, an equivalent of the variance, as; (sum of square differences between the projected samples and their class mean).

$$\widetilde{s}_i^2 = \sum_{y \in \omega_i} (y - \widetilde{\mu}_i)^2 \checkmark$$

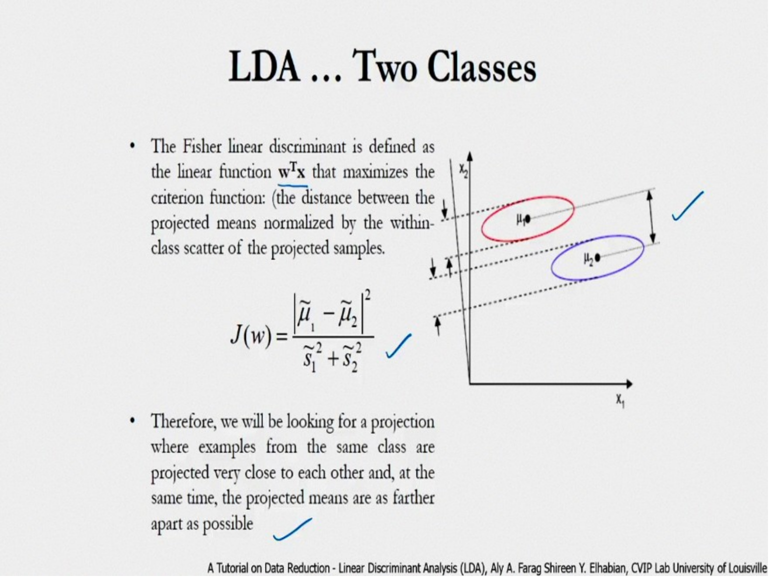
- $\widetilde{s}_i^2$ measures the variability within class ω_i after projecting it on the y-space.
- Thus $\widetilde{s}_1^2 + \widetilde{s}_2^2$ measures the variability within the two classes at hand after projection, hence it is called *within-class scatter* of the projected samples.

So, that is the solution of the, this problem is given by Fisher. So that is why, this method is called Fisher Linear Discriminant Analysis. So what is the solution of these problem? The solution of this problem is to maximize a function that represents, the difference between the means, normalized by a measure of within-class variability. So, that means, I am considering the information or the measure of the within-class variability; and I can consider as a scatter. For each class, we define the scatter and equivalent to the variance.

So, I can consider a, this is the scatter, and it is equivalent of the variance. That is the sum of square differences between the projected samples and their class mean. So, that I am considering. The sum of square differences between the projected samples and their class mean, I am considering. So, $\widetilde{s_i}^2$ measures the variability within the class w $_i$. After projecting it onto the y-space. So, y-space means it is the projected space. So, $\widetilde{s_i}^2$ means, it is a measure of the variability within the class, the class is w $_i$.

So, that means if I consider this one, $\widetilde{s_1}^2+\widetilde{s_2}^2$, that gives the measure of the variability within the 2 classes after the projection. So, that is called the within-class scatter of the projected sample. That means, I am considering $\widetilde{s_1}^2+\widetilde{s_2}^2$; that measures the variability within the 2 classes after the projection, and it is called the within-class scatter of the projected samples.

(Refer Slide Time: 21:46)



LDA ... Two Classes

- The Fisher linear discriminant is defined as the linear function $\mathbf{w}^T\mathbf{x}$ that maximizes the criterion function: (the distance between the projected means normalized by the within-class scatter of the projected samples.

$$J(w)=\frac{\left|\tilde{\mu}_1-\tilde{\mu}_2\right|^2}{\tilde{s}_1^2+\tilde{s}_2^2} \checkmark$$

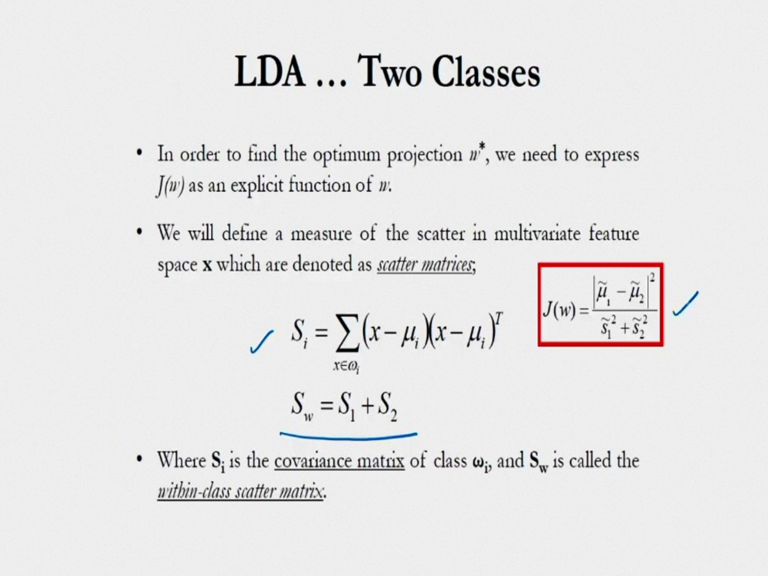
- Therefore, we will be looking for a projection where examples from the same class are projected very close to each other and, at the same time, the projected means are as farther apart as possible ✓

A Tutorial on Data Reduction - Linear Discriminant Analysis (LDA), Aly A. Farag Shireen Y. Elhabian, CVIP Lab University of Louisville

So, in case of the Fisher Linear Discriminant Function, we define a linear function, the linear function is w ^{T} x that maximize the criterion function. What is the criterion function? The distance between the projected means normalized by the within-class scatter of the projected samples. So, I am considering this, the criterion function J (w), I am considering. So, the objective is to maximize the criterion function, I have to maximize this. That means the distance between the projected means normalized by the within-class scatter of the projected samples, I have to maximize.

So, that is the criterion function, I am considering in case of the Fisher Linear Discriminant. That means, in case of the LDA, what actually we are considering? We are looking for a projection, where the samples of the same class are projected very close to each other, and at the same time the projected means are further apart as far as possible. So, that is I am considering. So, one is within-class distance, another one is the between-class distance; that is I am considering. And based on this, I am determining that projection direction. So, this concept I am showing here again. So, that means, the maximum separation between the classes, but samples from the same class are projected very close to each other. So, that I am also considering.

(Refer Slide Time: 23:15)



LDA … Two Classes

- In order to find the optimum projection w^*, we need to express $J(w)$ as an explicit function of w:
- We will define a measure of the scatter in multivariate feature space **x** which are denoted as *scatter matrices*;

$$\boxed{J(w)=\frac{\left|\tilde{\mu}_1-\tilde{\mu}_2\right|^2}{\tilde{s}_1^2+\tilde{s}_2^2}} \checkmark$$

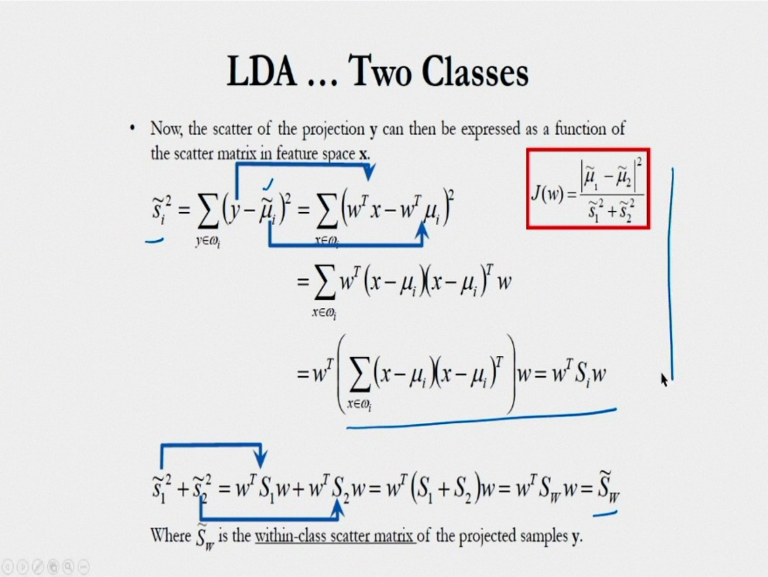
$$\checkmark \quad S_i=\sum_{x\in\omega_i}(x-\mu_i)(x-\mu_i)^T$$

$$\underline{S_w=S_1+S_2}$$

- Where $\mathbf{S}_i$ is the covariance matrix of class $\boldsymbol{\omega}_i$, and $\mathbf{S}_w$ is called the *within-class scatter matrix*.

In order to find the optimum projection w star, we need to express J (w) as an explicit function of w. So, I have to find the J (w), that is the criterion function. So, for this we are defining a measure of the scatter in multivariate feature space x, which is denoted as scatter matrix. So, I am considering S $_i$. So, S $_i$ is the covariance matrix of class w $_i$ and S $_W$ is called within-class scatter matrix. So, I can determine the within-class scatter matrix from S1 and S2. So, S1 is the covariance matrix of the class 1, and S2 is the covariance matrix of the class 2. So, from this I can determine S $_W$, that is nothing but within-class scatter matrix. This is important for considering these criterion function. So, J (w) I am considering.

(Refer Slide Time: 24:17)



LDA … Two Classes

- Now, the scatter of the projection **y** can then be expressed as a function of the scatter matrix in feature space **x**.

$$J(w)=\frac{\left|\tilde{\mu}_1-\tilde{\mu}_2\right|^2}{\tilde{s}_1^2+\tilde{s}_2^2}$$

$$\tilde{s}_i^2=\sum_{y\in\omega_i}\left(y-\tilde{\mu}_i\right)^2=\sum_{x\in\omega_i}\left(w^Tx-w^T\mu_i\right)^2$$

$$=\sum_{x\in\omega_i}w^T\left(x-\mu_i\right)\left(x-\mu_i\right)^Tw$$

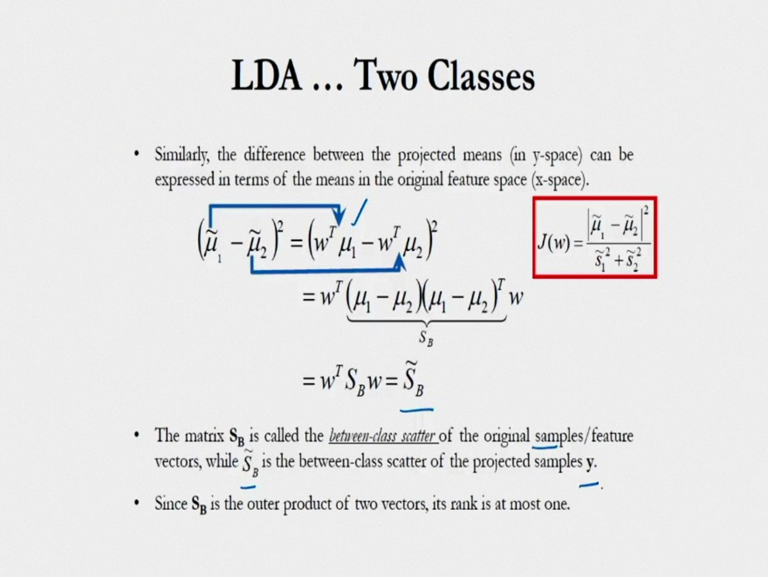
$$=w^T\left(\sum_{x\in\omega_i}\left(x-\mu_i\right)\left(x-\mu_i\right)^T\right)w=w^TS_iw$$

$$\tilde{s}_1^2+\tilde{s}_2^2=w^TS_1w+w^TS_2w=w^T\left(S_1+S_2\right)w=w^TS_Ww=\tilde{S}_W$$

Where $\tilde{S}_W$ is the within-class scatter matrix of the projected samples **y**.

Now, the scatter of the projection y can be expressed as a function of the scatter matrix in feature space x. So, here you can see, I am considering the projected data. So, $\tilde{s}_i^2$, that I can determine. So, y is the projected data. And you know, what is $\tilde{\mu}_i$; you know. So, from this you can determine this one, just you can see this one. And similarly, you can also determine the S2. S1 and S2, that is the $\tilde{s}_1^2$ and $\tilde{s}_2^2$, that you can determine. And that is nothing but S $_W$. So, $\tilde{S}_W$ is the within-class scatter matrix of the projected sample y. So, you can see the mathematics, and this derivation you can see. So this is a very simple derivation.

(Refer Slide Time: 25: 07)



LDA … Two Classes

- Similarly, the difference between the projected means (in y-space) can be expressed in terms of the means in the original feature space (x-space).

$$(\tilde{\mu}_1 - \tilde{\mu}_2)^2 = (w^T\mu_1 - w^T\mu_2)^2$$

$$= w^T \underbrace{(\mu_1 - \mu_2)(\mu_1 - \mu_2)^T}_{S_B} w$$

$$= w^T S_B w = \tilde{S}_B$$

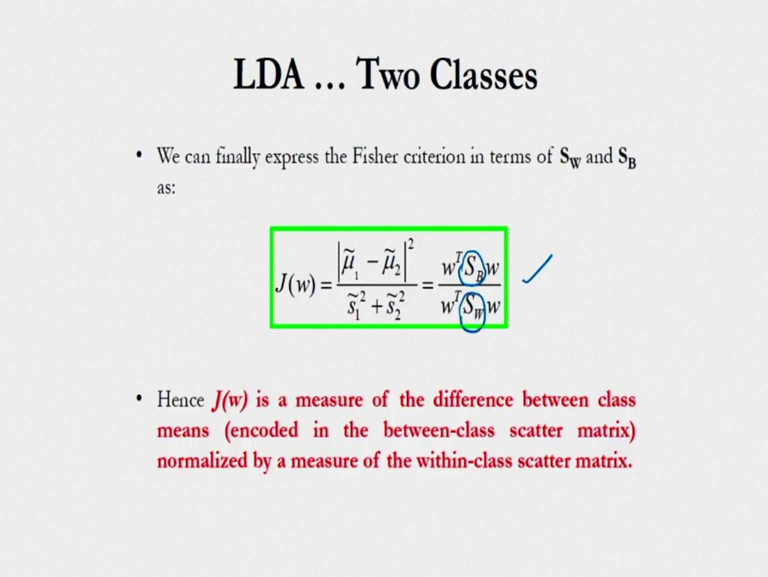
$$\boxed{J(w) = \frac{|\tilde{\mu}_1 - \tilde{\mu}_2|^2}{\tilde{s}_1^2 + \tilde{s}_2^2}}$$

- The matrix $\mathbf{S_B}$ is called the *between-class scatter* of the original samples/feature vectors, while $\tilde{S}_B$ is the between-class scatter of the projected samples **y**.
- Since $\mathbf{S_B}$ is the outer product of two vectors, its rank is at most one.

And based on this you can see, because I am considering the projected means, the projected mean is $\widetilde{\mu_1}$and the $\widetilde{\mu_2}$. So, the separation between these 2 means, that is the projected mean should be maximum. So, just I am determining this w ^{T} μ_1, that corresponds to $\widetilde{\mu_1}$, and $\widetilde{\mu_2}$ is nothing but w ^{T} μ_2. So, you know this expression, and from this you can see, I am getting $\widetilde{S_B}$.

So, S $_B$ is nothing but the between-class scatter matrix. So, you can see how to determine the within-class scatter matrix S $_W$ and also, we can determine the between-class scatter matrix S $_B$. So, S $_B$ is the between-class scatter of the original samples. And what is $\widetilde{S_B}$, that is the between-class scatter of the projected sample y, that you can determine.

(Refer Slide Time: 26:01)



LDA … Two Classes

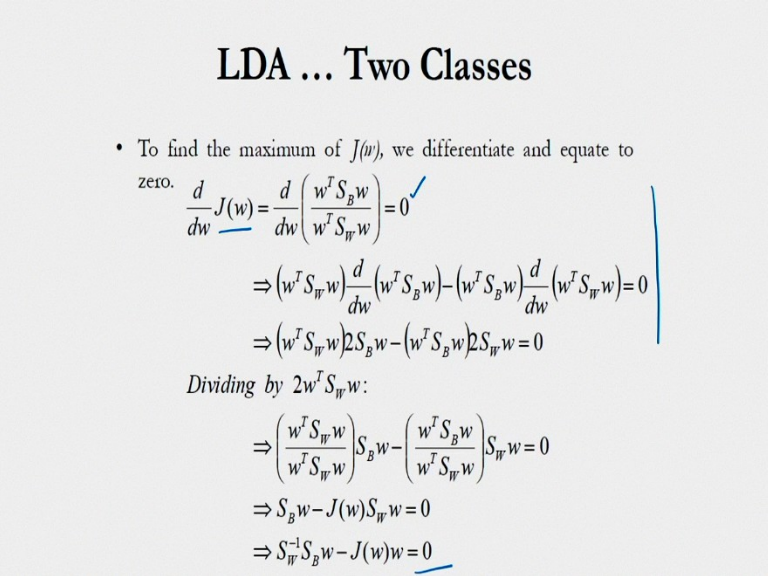
- We can finally express the Fisher criterion in terms of $\mathbf{S_W}$ and $\mathbf{S_B}$ as:

$$\boxed{J(w) = \frac{\left|\tilde{\mu}_1 - \tilde{\mu}_2\right|^2}{\tilde{s}_1^2 + \tilde{s}_2^2} = \frac{w^T S_B w}{w^T S_W w}} \checkmark$$

- Hence ***J(w)* is a measure of the difference between class means (encoded in the between-class scatter matrix) normalized by a measure of the within-class scatter matrix.**

And after this, this Fisher criterion function, that is J (w) can be expressed in terms of the between-class scatter matrix and the within-class scatter matrix, that can be represented like this. So, J (w) is nothing but $w^T S_B w$ divided by $w^T S_W w$. So, it can be represented like this. So, J (w) is the major of the difference between-class means, that is encoded in the between-class scatter matrix normalized by a measure of within-class scatter matrix. So, here you can see. So, it is the between-class scatter matrix and it is normalized by a measure of the within-class scatter matrix. So, we did as the S_W.

(Refer Slide Time: 26:46)



LDA … Two Classes

- To find the maximum of $J(w)$, we differentiate and equate to zero. $\frac{d}{dw}\underline{J(w)} = \frac{d}{dw}\left(\frac{w^T S_B w}{w^T S_W w}\right) = 0 \checkmark$

$$\Rightarrow \left(w^T S_W w\right)\frac{d}{dw}\left(w^T S_B w\right) - \left(w^T S_B w\right)\frac{d}{dw}\left(w^T S_W w\right) = 0$$

$$\Rightarrow \left(w^T S_W w\right)2S_B w - \left(w^T S_B w\right)2S_W w = 0$$

Dividing by $2w^T S_W w$:

$$\Rightarrow \left(\frac{w^T S_W w}{w^T S_W w}\right)S_B w - \left(\frac{w^T S_B w}{w^T S_W w}\right)S_W w = 0$$

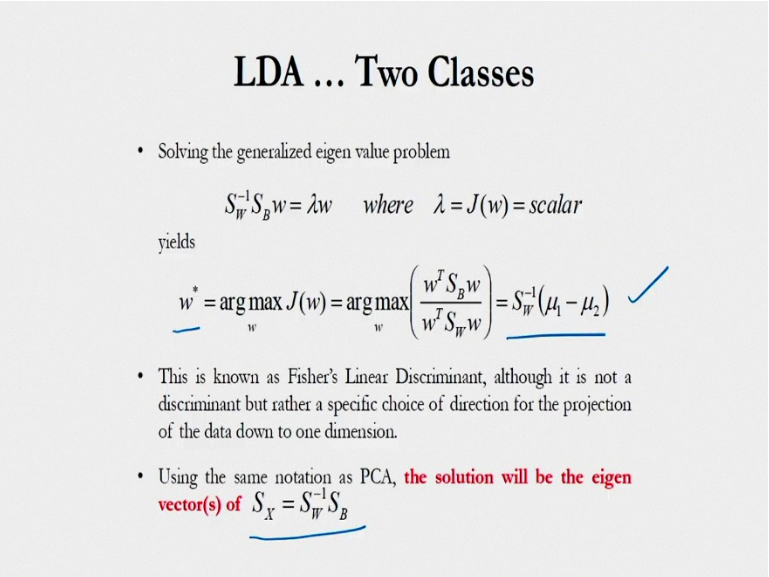
$$\Rightarrow S_B w - J(w)S_W w = 0$$

$$\Rightarrow S_W^{-1}S_B w - J(w)w = \underline{0}$$

And I have to maximize this criterion function J (w), so that is why I am taking the differentiation with respect to w. So, w is the projection vector. So, I have to maximize J (w), with respect to w. So, that is why I am doing the differentiation. You can see this mathematics. So, how to do the differentiation, by using the chain rule. So, you can do the differentiation by using the chain rule, and since I have to find the maximum value; so that is why, I am equating it to 0.

So, I have to find the maximum of J (w). After doing all this mathematics, I will be getting this one. So, you see this mathematics, mainly just I am applying the differentiation, applying the chain rule and just equating it to 0, because I have to find the maximum of J (w).

(Refer Slide Time: 27:38)



LDA ... Two Classes

- Solving the generalized eigen value problem

$$S_W^{-1} S_B w = \lambda w \quad where \quad \lambda = J(w) = scalar$$

yields

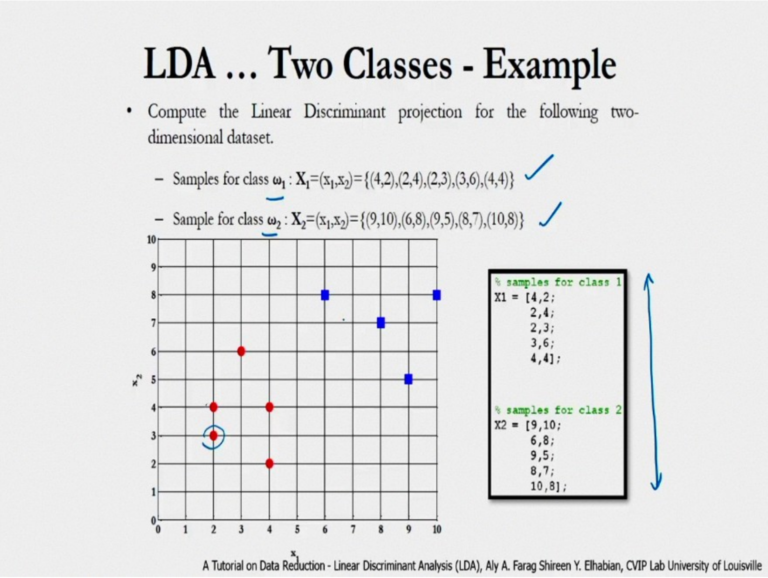
$$w^* = \underset{w}{\arg\max}\, J(w) = \underset{w}{\arg\max} \left(\frac{w^T S_B w}{w^T S_W w} \right) = S_W^{-1}(\mu_1 - \mu_2)$$

- This is known as Fisher's Linear Discriminant, although it is not a discriminant but rather a specific choice of direction for the projection of the data down to one dimension.
- Using the same notation as PCA, **the solution will be the eigen vector(s) of** $S_X = S_W^{-1} S_B$

So, it is nothing but, the solving the generalized eigenvalue problem. So, $S_W{}^{-1}$, that is the inverse of the within-class scatter matrix, into S $_B$, the between-class scatter matrix, and w is the projection vector, λ w. So, λ is the eigenvalues. So, λ is the scalar. So, corresponding to this, if I consider this eigenvalue problem, I can determine the vector w. That I can determine, that is the projection vector; I can determine. So, this w star, I can determine that is nothing but $S_W{}^{-1}(\mu_1 - \mu_2)$. So, here you can see I am determining the best projection direction, the optimum projection direction w star. So, this is known as Fisher's Linear Discriminant.

And if I consider the same notation as PCA, the solution will be the eigenvectors of S $_X$, because in case of the PCA also, we determine the eigenvectors of the covariance matrix. So, similarly the solution will be the eigenvectors of the S $_X$. So, that is nothing but $S_W{}^{-1}$ into S $_B$. So, this is the very similar to the PCA, the Principal Component Analysis. In PCA, we consider the eigenvectors of the covariance matrix. In this case you can see, I am considering $S_W{}^{-1}$ into S $_B$. Also, one is the within-class scatter matrix, another one is the between-class scatter matrix.

(Refer Slide Time: 29:19)



LDA … Two Classes - Example

- Compute the Linear Discriminant projection for the following two-dimensional dataset.
  - Samples for class ω_1 : $X_1=(x_1,x_2)=\{(4,2),(2,4),(2,3),(3,6),(4,4)\}$
  - Sample for class ω_2 : $X_2=(x_1,x_2)=\{(9,10),(6,8),(9,5),(8,7),(10,8)\}$

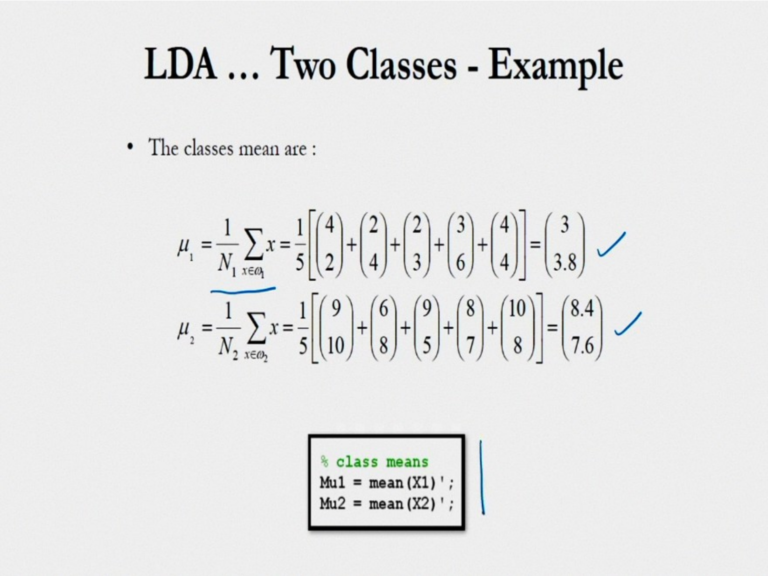
```
% samples for class 1
X1 = [4,2;
      2,4;
      2,3;
      3,6;
      4,4];

% samples for class 2
X2 = [9,10;
      6,8;
      9,5;
      8,7;
      10,8];
```

A Tutorial on Data Reduction - Linear Discriminant Analysis (LDA), Aly A. Farag Shireen Y. Elhabian, CVIP Lab University of Louisville

So, I am considering one numerical example. So, how to apply the LDA for 2 classes. So, I am considering samples of the class w1. So, I am considering 2 classes, w1 and w2, and I am considering the samples for the class w1. So, these are the samples. The samples are 2-dimensional. And, similarly I am considering the samples of the class w2. That is also 2-dimensional, and I am showing the Matlab code for this. And, I am considering the samples X1 and X2 corresponding to the classes w1 and w2 respectively.

And you can see the, you can plot the samples corresponding to these 2 classes. One is the green sample; you can see the green colored sample. And another one is the blue colored samples.

(Refer Slide Time: 30:08)



LDA ... Two Classes - Example

- The classes mean are :

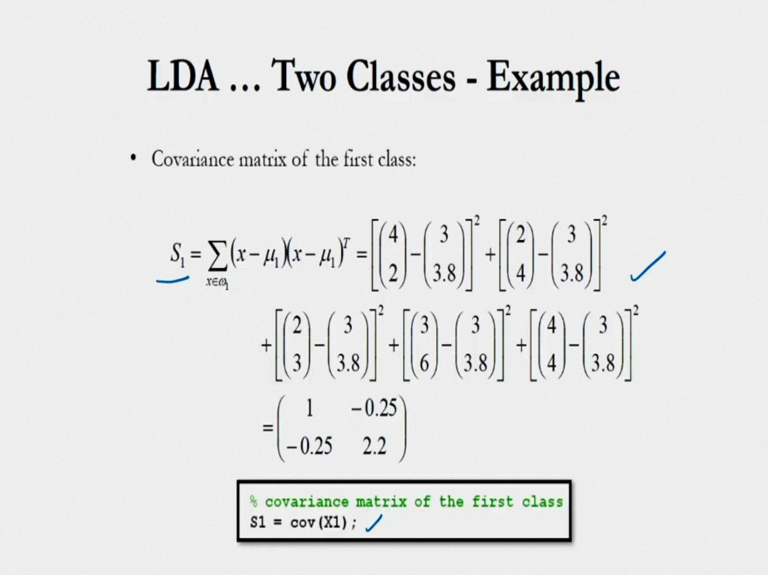
$$\mu_1 = \underline{\frac{1}{N_1}\sum_{x\in\omega_1}} x = \frac{1}{5}\left[\begin{pmatrix}4\\2\end{pmatrix}+\begin{pmatrix}2\\4\end{pmatrix}+\begin{pmatrix}2\\3\end{pmatrix}+\begin{pmatrix}3\\6\end{pmatrix}+\begin{pmatrix}4\\4\end{pmatrix}\right]=\begin{pmatrix}3\\3.8\end{pmatrix} \checkmark$$

$$\mu_2 = \frac{1}{N_2}\sum_{x\in\omega_2} x = \frac{1}{5}\left[\begin{pmatrix}9\\10\end{pmatrix}+\begin{pmatrix}6\\8\end{pmatrix}+\begin{pmatrix}9\\5\end{pmatrix}+\begin{pmatrix}8\\7\end{pmatrix}+\begin{pmatrix}10\\8\end{pmatrix}\right]=\begin{pmatrix}8.4\\7.6\end{pmatrix} \checkmark$$

```
% class means
Mu1 = mean(X1)';
Mu2 = mean(X2)';
```

After this, this, the class means I can determine by using this expression. So, corresponding to the class w1, I can determine the mean of the samples. And similarly, corresponding to the second class also I can determine the class mean. So, that I can determine. And in the Matlab you can write like this. You can determine the mean of X1, and also the mean of X2 you can determine.

(Refer Slide Time: 30:31)



LDA ... Two Classes - Example

- Covariance matrix of the first class:

$$\underline{S_1} = \sum_{x\in\omega_1}(x-\mu_1)(x-\mu_1)^T = \left[\begin{pmatrix}4\\2\end{pmatrix}-\begin{pmatrix}3\\3.8\end{pmatrix}\right]^2+\left[\begin{pmatrix}2\\4\end{pmatrix}-\begin{pmatrix}3\\3.8\end{pmatrix}\right]^2 \checkmark$$

$$+\left[\begin{pmatrix}2\\3\end{pmatrix}-\begin{pmatrix}3\\3.8\end{pmatrix}\right]^2+\left[\begin{pmatrix}3\\6\end{pmatrix}-\begin{pmatrix}3\\3.8\end{pmatrix}\right]^2+\left[\begin{pmatrix}4\\4\end{pmatrix}-\begin{pmatrix}3\\3.8\end{pmatrix}\right]^2$$

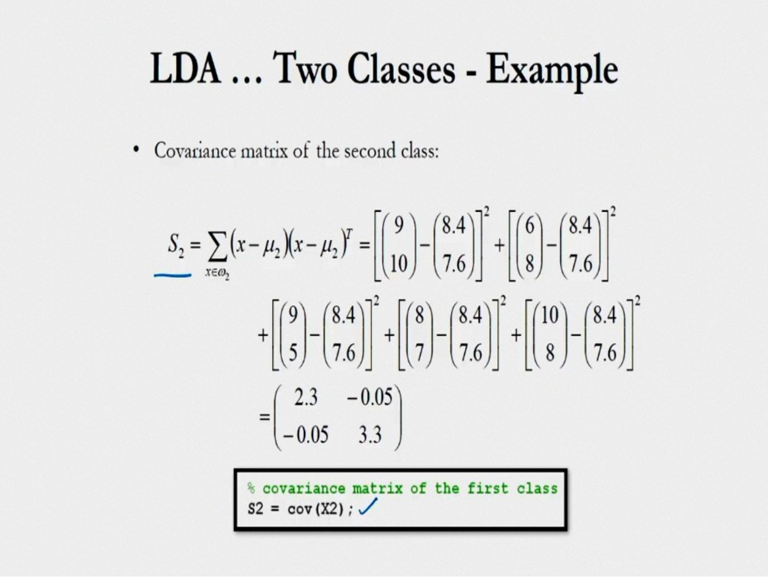
$$=\begin{pmatrix}1 & -0.25\\ -0.25 & 2.2\end{pmatrix}$$

```
% covariance matrix of the first class
S1 = cov(X1); ✓
```

After this, the covariance matrix of the first class also, you can determine. That is nothing but S1. So, S1 you can determine, that is nothing but the covariance matrix of the first class. And in a

Matlab, you have to right simply the covariance of X1. So, you can write like this. So, you can determine S1.

(Refer Slide Time: 30:50)



LDA … Two Classes - Example

- Covariance matrix of the second class:

$$S_2 = \sum_{x \in \omega_2} (x-\mu_2)(x-\mu_2)^T = \left[\begin{pmatrix} 9 \\ 10 \end{pmatrix} - \begin{pmatrix} 8.4 \\ 7.6 \end{pmatrix}\right]^2 + \left[\begin{pmatrix} 6 \\ 8 \end{pmatrix} - \begin{pmatrix} 8.4 \\ 7.6 \end{pmatrix}\right]^2$$

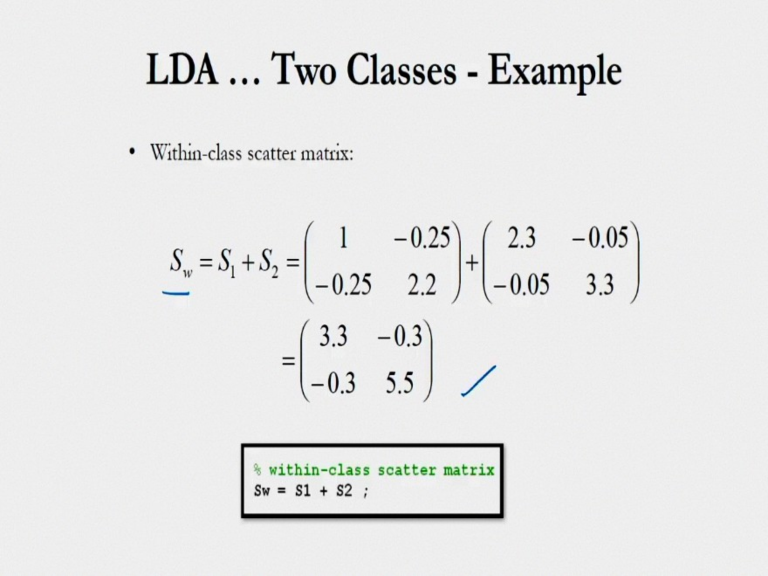
$$+ \left[\begin{pmatrix} 9 \\ 5 \end{pmatrix} - \begin{pmatrix} 8.4 \\ 7.6 \end{pmatrix}\right]^2 + \left[\begin{pmatrix} 8 \\ 7 \end{pmatrix} - \begin{pmatrix} 8.4 \\ 7.6 \end{pmatrix}\right]^2 + \left[\begin{pmatrix} 10 \\ 8 \end{pmatrix} - \begin{pmatrix} 8.4 \\ 7.6 \end{pmatrix}\right]^2$$

$$= \begin{pmatrix} 2.3 & -0.05 \\ -0.05 & 3.3 \end{pmatrix}$$

```
% covariance matrix of the first class
S2 = cov(X2);
```

And similarly, you can determine the covariance matrix of the second class. So, S2 is the covariance matrix of the second class, you can determine. And in the Matlab S2 is equal to covariance as X2 that you can determine.

(Refer Slide Time: 31:03)



LDA … Two Classes - Example

- Within-class scatter matrix:

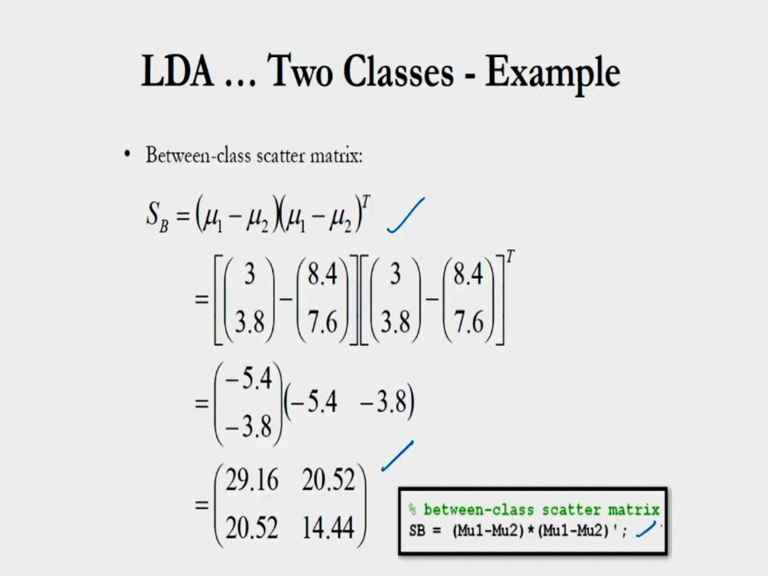
$$S_w = S_1 + S_2 = \begin{pmatrix} 1 & -0.25 \\ -0.25 & 2.2 \end{pmatrix} + \begin{pmatrix} 2.3 & -0.05 \\ -0.05 & 3.3 \end{pmatrix}$$

$$= \begin{pmatrix} 3.3 & -0.3 \\ -0.3 & 5.5 \end{pmatrix}$$

```
% within-class scatter matrix
Sw = S1 + S2 ;
```

And from this S1 and S2, you can determine within-class scatter matrix S W, you can determine. That is the within-class scatter matrix, you can determine from S1 and S2. So, you will be getting this.

(Refer Slide Time: 31:17)



LDA … Two Classes - Example

- Between-class scatter matrix:

$$S_B = (\mu_1 - \mu_2)(\mu_1 - \mu_2)^T$$

$$= \left[\begin{pmatrix}3\\3.8\end{pmatrix} - \begin{pmatrix}8.4\\7.6\end{pmatrix}\right]\left[\begin{pmatrix}3\\3.8\end{pmatrix} - \begin{pmatrix}8.4\\7.6\end{pmatrix}\right]^T$$

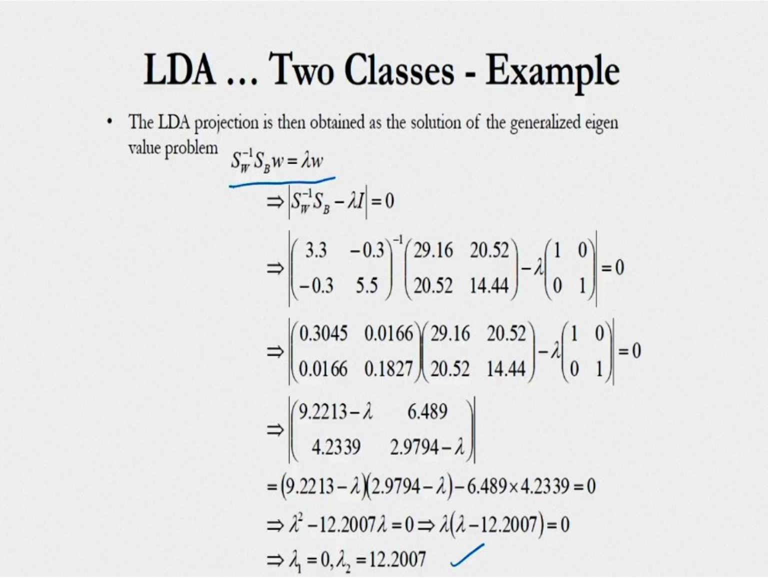
$$= \begin{pmatrix}-5.4\\-3.8\end{pmatrix}\begin{pmatrix}-5.4 & -3.8\end{pmatrix}$$

$$= \begin{pmatrix}29.16 & 20.52\\20.52 & 14.44\end{pmatrix}$$

```
% between-class scatter matrix
SB = (Mu1-Mu2)*(Mu1-Mu2)';
```

And after this, you can also determine the between-class scatter matrix from these 2 means, because already you have calculated μ_1 and μ_2. So, from you μ_1 and μ_2 you can determine between-class scatter matrix. So, you can see. So, I am computing the between-class scatter matrix, and even in the Matlab also it is very simple. So, you can determine between-class scatter matrix.

(Refer Slide Time: 31:41)



LDA … Two Classes - Example

- The LDA projection is then obtained as the solution of the generalized eigen value problem $S_W^{-1}S_B w = \lambda w$

$$\Rightarrow \left|S_W^{-1}S_B - \lambda I\right| = 0$$

$$\Rightarrow \left|\begin{pmatrix} 3.3 & -0.3 \\ -0.3 & 5.5 \end{pmatrix}^{-1}\begin{pmatrix} 29.16 & 20.52 \\ 20.52 & 14.44 \end{pmatrix} - \lambda\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}\right| = 0$$

$$\Rightarrow \left|\begin{pmatrix} 0.3045 & 0.0166 \\ 0.0166 & 0.1827 \end{pmatrix}\begin{pmatrix} 29.16 & 20.52 \\ 20.52 & 14.44 \end{pmatrix} - \lambda\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}\right| = 0$$

$$\Rightarrow \left|\begin{pmatrix} 9.2213-\lambda & 6.489 \\ 4.2339 & 2.9794-\lambda \end{pmatrix}\right|$$

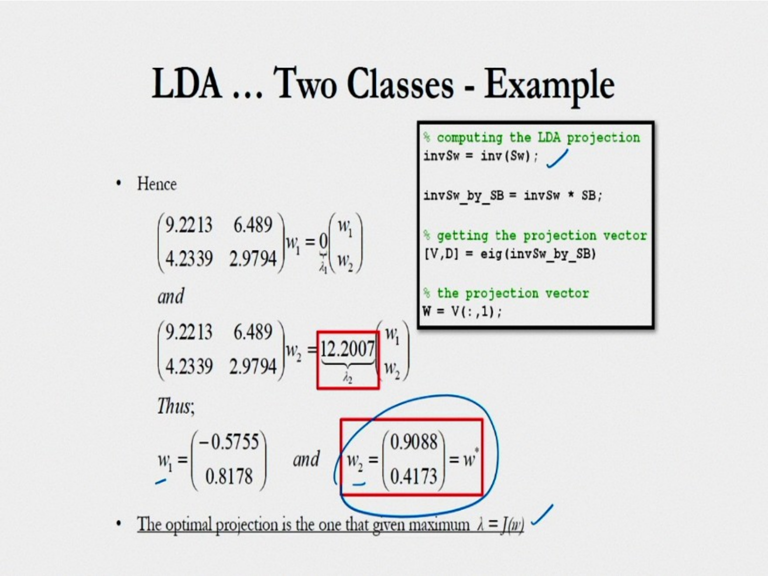
$$= (9.2213-\lambda)(2.9794-\lambda) - 6.489 \times 4.2339 = 0$$

$$\Rightarrow \lambda^2 - 12.2007\lambda = 0 \Rightarrow \lambda(\lambda - 12.2007) = 0$$

$$\Rightarrow \lambda_1 = 0, \lambda_2 = 12.2007$$

After this, the problem is the eigenvalue problem, that you have to solve. So, the eigenvalue problem is this. So, lambda is the eigenvalue. So, this eigenvalue problem, you can solve like this. And I will be getting the, I will be getting 2 eigenvalues, λ_1 and λ_2. So, λ_1 is 0 and λ_2 is 12.2007. So, you will be getting 2 eigenvalues. This is nothing but the solution of the generalized eigenvalue problem. So, you can solve this problem.

(Refer Slide Time: 32:12)



LDA … Two Classes - Example

- Hence

$$\begin{pmatrix} 9.2213 & 6.489 \\ 4.2339 & 2.9794 \end{pmatrix} w_1 = \underbrace{0}_{\lambda_1}\begin{pmatrix} w_1 \\ w_2 \end{pmatrix}$$

and

$$\begin{pmatrix} 9.2213 & 6.489 \\ 4.2339 & 2.9794 \end{pmatrix} w_2 = \boxed{\underbrace{12.2007}_{\lambda_2}}\begin{pmatrix} w_1 \\ w_2 \end{pmatrix}$$

Thus;

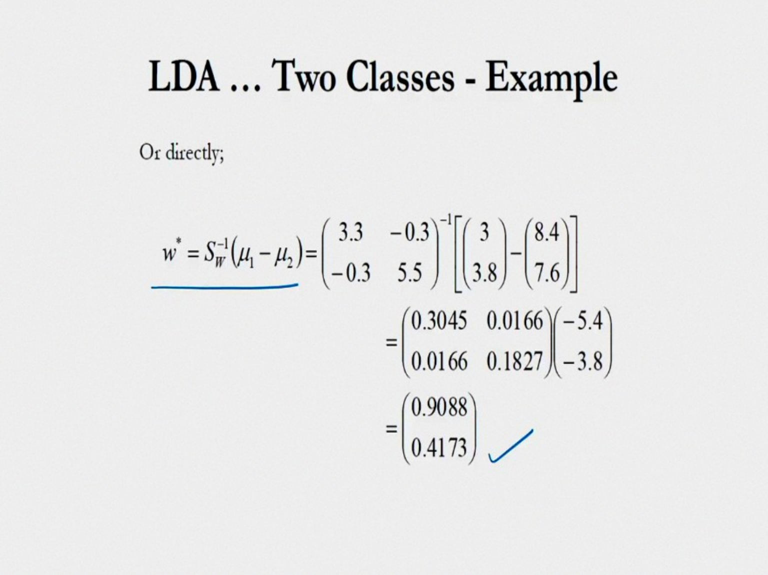
$$w_1 = \begin{pmatrix} -0.5755 \\ 0.8178 \end{pmatrix} \quad and \quad \boxed{w_2 = \begin{pmatrix} 0.9088 \\ 0.4173 \end{pmatrix} = w^*}$$

- The optimal projection is the one that given maximum $\lambda = J(w)$

```
% computing the LDA projection
invSw = inv(Sw);

invSw_by_SB = invSw * SB;

% getting the projection vector
[V,D] = eig(invSw_by_SB)

% the projection vector
W = V(:,1);
```

And after this, I can determine the vector w. So, you can determine w1 and w2; you can determine. So, we can compute the LDA projection. So, it is nothing but, in the Matlab you can do like this. So, you have to find the inverse of S_W and, I will be getting the projection vector; w is the projection vector. So, I will be getting w1 and w2. And which one is the optimum projection direction corresponding to LDA? That is the, w2 is the optimum projection direction that I can determine; which gives maximum J (w). So, this w2 is the optimum projection vector, that I can determine, because it gives maximum J (w).

(Refer Slide Time: 33:03)



LDA ... Two Classes - Example

Or directly;

$$\underline{w^* = S_W^{-1}(\mu_1 - \mu_2)} = \begin{pmatrix} 3.3 & -0.3 \\ -0.3 & 5.5 \end{pmatrix}^{-1} \left[\begin{pmatrix} 3 \\ 3.8 \end{pmatrix} - \begin{pmatrix} 8.4 \\ 7.6 \end{pmatrix} \right]$$

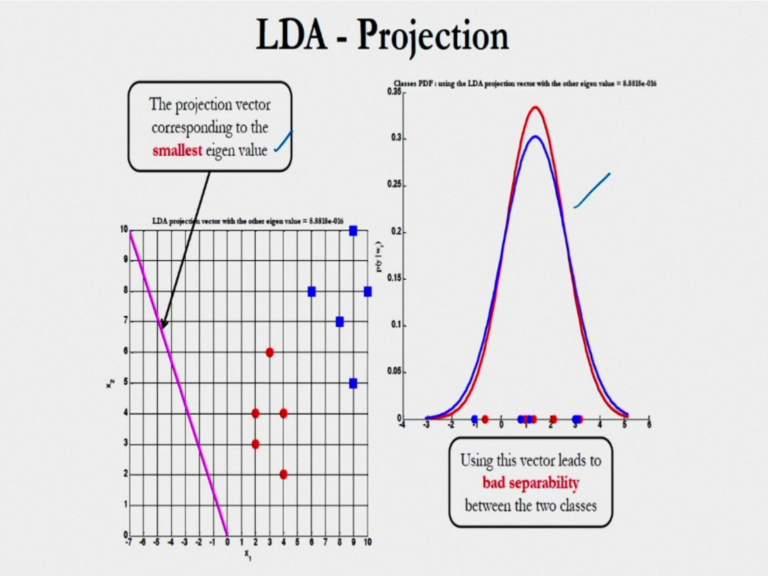
$$= \begin{pmatrix} 0.3045 & 0.0166 \\ 0.0166 & 0.1827 \end{pmatrix} \begin{pmatrix} -5.4 \\ -3.8 \end{pmatrix}$$

$$= \begin{pmatrix} 0.9088 \\ 0.4173 \end{pmatrix} \checkmark$$

Or maybe directly, we can compute like this. The optimum projection direction order vector; I can determine like this. So, $S_W{}^{-1}$, that is nothing but the inverse of the within-class scatter matrix, $\mu_1 - \mu_2$. So, from this you can determine the optimum projection vector that you can determine. So, this is one example. This LDA you can apply for C number of classes. So, in the book you can get this information, how to apply the LDA for C number of classes.

But, in my discussion I only considered 2 classes. So, how to apply the LDA for 2 classes. For this, you have to determine S_W and S_B; one is the within-class scatter matrix, another one is the between-class scatter matrix.

(Refer Slide Time: 33:52)



LDA - Projection

The projection vector corresponding to the **smallest** eigen value

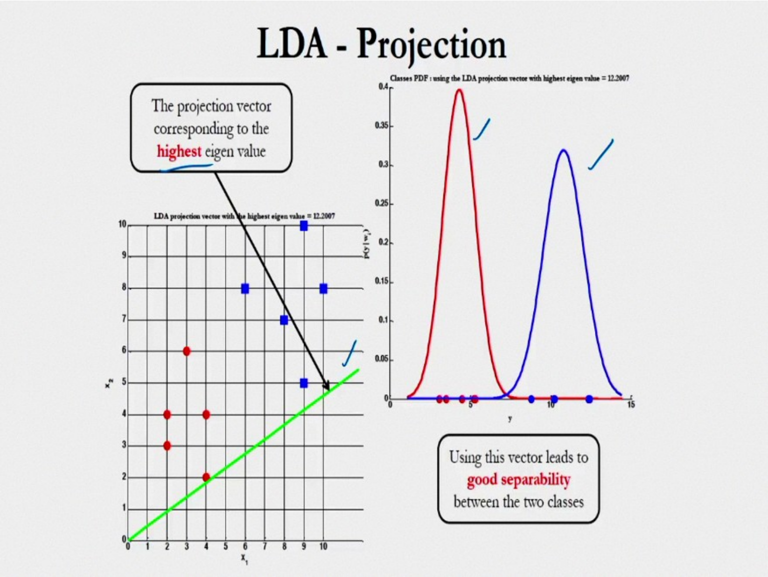
Using this vector leads to **bad separability** between the two classes

And here you can see, I am showing one projection direction corresponding to smallest eigenvalue. So, smallest eigenvalue I am considering, and I am showing the projection direction. And in this case, if I considered the PDF of the classes, they are not well separated. That means, that there is no discrimination between the classes, corresponding to this projection direction. In case of the PCA also, we considered the eigenvalues and the corresponding eigenvectors.

So, in case of the PCA, we consider the highest eigenvalue and corresponding eigenvectors, that we considered. And if I consider, the smallest eigenvalue that corresponds to the redundant information, or maybe the noise; that we can neglect in case of the PCA. Here in case of LDA, what I am considering, the smallest eigenvalue, I am considering. And corresponding to this, I can determine the projection direction.

And here you can see, corresponding to the smallest eigenvalue, the separation between the 2 classes is not maximum. It is overlapping; overlapping of the PDF of the classes, that is the bad separability.

(Refer Slide Time: 35:03)



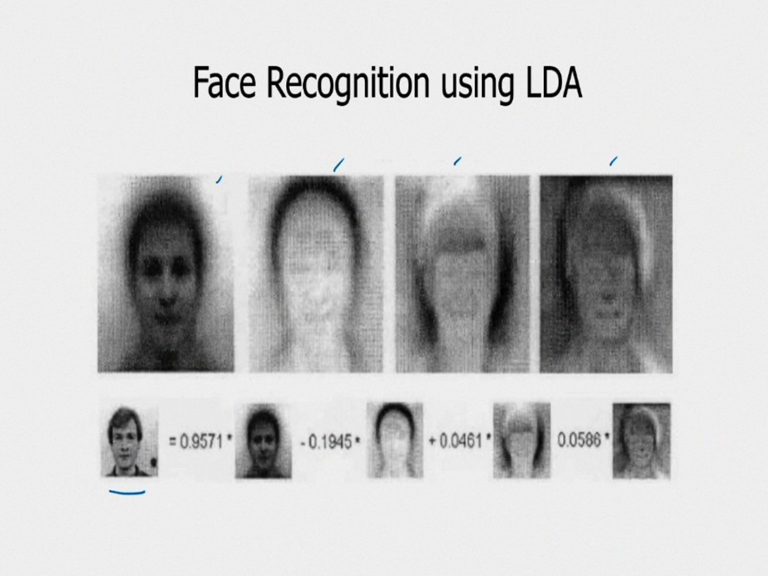
LDA - Projection

The projection vector corresponding to the **highest** eigen value

Using this vector leads to **good separability** between the two classes

But if I consider, the highest eigenvalue; and corresponding to this, I can determine the projection direction. In this case, you can get the good separability between the classes. So, you can see, I am showing the PDF of the classes, and you can find the good separability between the classes corresponding to the highest eigenvalue, you can see.

(Refer Slide Time: 35:25)

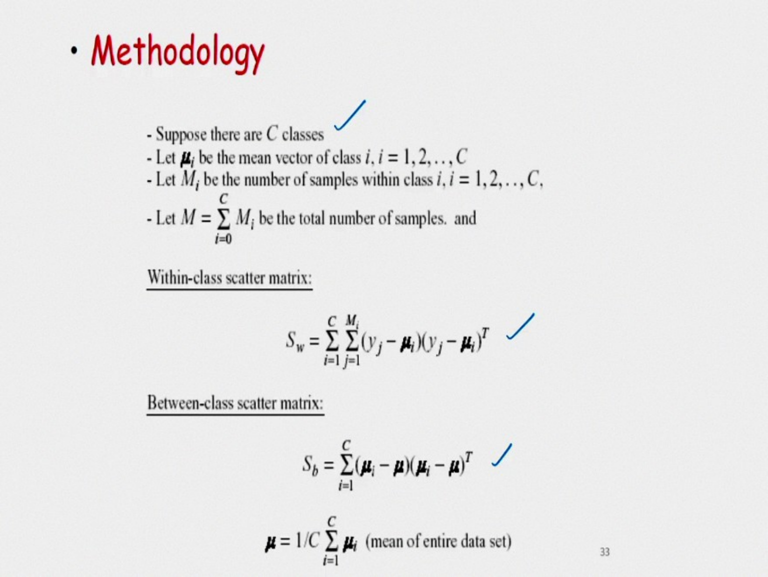


Face Recognition using LDA

Now in case of the PCA, we have seen how to recognize a particular face. The face recognition using PCA, that concept already I have explained. That means, any face can be expressed as a linear combination of eigenfaces. So, you can see, I am considering the eigenfaces like this. So, in my class, I have explained how to determine the eigenfaces. And the any face can be represented by a linear combination of the eigenfaces.

Similarly, in case of the LDA, any face can be represented by a linear combination of Fisherfaces. In case of the PCA, we consider the eigenfaces. But in this LDA, we are considering Fisherfaces. So, that means, the any face can be represented by linear combination of Fisherfaces. So, that concept I am going to explain now.

(Refer Slide Time: 36:22)



- Methodology

- Suppose there are C classes ✓
- Let $\boldsymbol{\mu}_i$ be the mean vector of class i, $i = 1, 2, \ldots, C$
- Let M_i be the number of samples within class i, $i = 1, 2, \ldots, C$,
- Let $M = \sum_{i=0}^{C} M_i$ be the total number of samples. and

Within-class scatter matrix:

$$S_w = \sum_{i=1}^{C} \sum_{j=1}^{M_i} (y_j - \boldsymbol{\mu}_i)(y_j - \boldsymbol{\mu}_i)^T \checkmark$$

Between-class scatter matrix:

$$S_b = \sum_{i=1}^{C} (\boldsymbol{\mu}_i - \boldsymbol{\mu})(\boldsymbol{\mu}_i - \boldsymbol{\mu})^T \checkmark$$

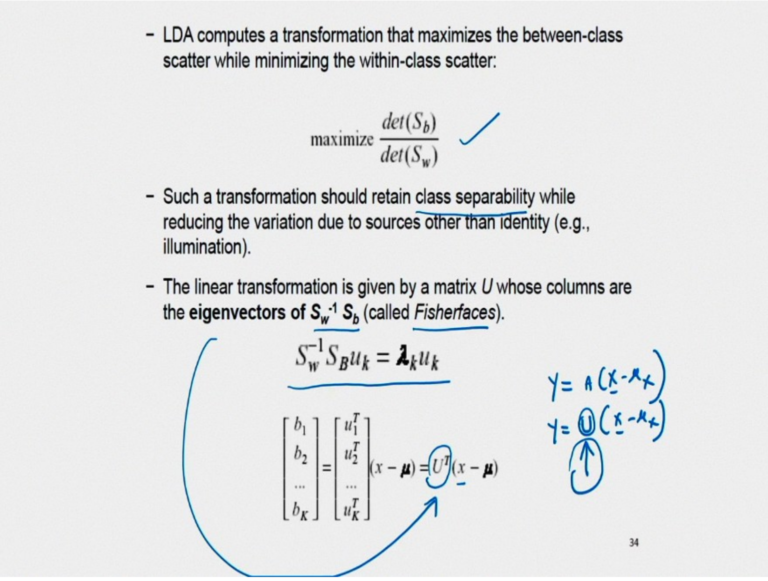
$$\boldsymbol{\mu} = 1/C \sum_{i=1}^{C} \boldsymbol{\mu}_i \text{ (mean of entire data set)}$$

33

So, how to recognize a particular face? Suppose, we have C number of classes. This μ_i, I can determine; that is the mean vector of the class, I can determine. So, I have C number of classes. So, also I am considering M_i number of samples for the classes, within a particular class. So, from this, I can determine that, what is the total number of samples. The total number of sample is M, is equal to summation over 0 to C M_i. So, M_i be the number of samples within class i; that I am considering.

And from this, I can determine total number of samples, I can determine. And I have already explained, I can determine the within-class scatter matrix, and also I can determine the between-class scatter matrix from the input samples.

(Refer Slide Time: 37:15)



- LDA computes a transformation that maximizes the between-class scatter while minimizing the within-class scatter:

$$\text{maximize} \ \frac{det(S_b)}{det(S_w)}$$

- Such a transformation should retain class separability while reducing the variation due to sources other than identity (e.g., illumination).
- The linear transformation is given by a matrix *U* whose columns are the **eigenvectors of $S_w^{-1} S_b$** (called *Fisherfaces*).

$$S_W^{-1} S_B u_k = \lambda_k u_k$$

$$\begin{bmatrix} b_1 \\ b_2 \\ \dots \\ b_K \end{bmatrix} = \begin{bmatrix} u_1^T \\ u_2^T \\ \dots \\ u_K^T \end{bmatrix} (x - \boldsymbol{\mu}) = U^T (x - \boldsymbol{\mu})$$

$Y = A(\underline{x} - \mu_x)$

$Y = U(\underline{x} - \mu_x)$

34

After this, I am considering the criterion function; that function I am considering. So, what is the condition? I have to maximize the between-class scatter, but I have to minimize the within-class scatter, that is the condition. Because, I have to find the best projection direction, I have to find. So for this, what I have to do? I have to maximize the between-class scatter, and also I have to minimize the within-class scatter.

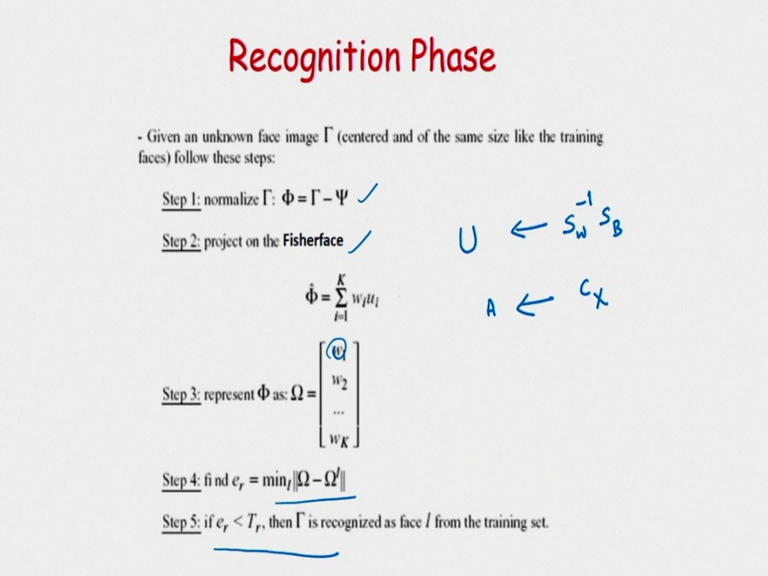
So, such a transformation should retain class separability, while reducing the variation due to source other than the identity. So, maybe the variation, maybe the illumination variation; I may consider. But, the main important point is the class separability. So, we have to find the class separability; that it is a maximum discrimination between the classes, I have to find. And already, I have explained that is nothing but the eigenvalue problem. So, this solution is something like this.

And, I will be getting the Fisherfaces, I will be getting. If I consider the eigenvectors of $S_W{}^{-1}$ into S $_B$. So, I will be getting the eigenvectors, that is nothing but the Fisherfaces. That means, you can see the projected data can be represented by a linear combination of the Fisherfaces. So here, you can see, I am considering, the U is the transform matrix and how it is obtained. It is nothing but the eigenvectors, I am considering. The eigenvectors of $S_W{}^{-1}$ into S $_B$. So, eigenvectors I am considering.

And based on this eigenvector, I can construct the transformation matrix. So, x is the input data minus mu; that is the mean is subtracted from the input data, that means the input data is normalized. And after this, I am considering the transformation. The transformation is the b is equal to U T x minus mu. So, suppose in case of the KL transformation, what I have considered Y is equal to A x minus mu x, I considered like this.

Similarly in this case, I am considering the U is the transformation matrix Y is this, and x minus mu x, like this; I am considering. So, this U is the transformation matrix. And this transformation matrix, I can obtain from the eigenvectors of S_W^{-1}; that is the inverse of the within-class scatter matrix into S $_{b.}$ So, I can get this one, this one U. That means any face can be represented by a linear combination of the Fisherfaces.

(Refer Slide Time: 39:57)



Recognition Phase

- Given an unknown face image Γ (centered and of the same size like the training faces) follow these steps:

Step 1: normalize Γ: $\Phi = \Gamma - \Psi$ ✓

Step 2: project on the **Fisherface** ✓

$$\hat{\Phi} = \sum_{i=1}^{K} w_i u_i$$

$U \leftarrow S_W^{-1} S_B$

$A \leftarrow C_X$

Step 3: represent Φ as: $\Omega = \begin{bmatrix} w_1 \\ w_2 \\ \dots \\ w_K \end{bmatrix}$

Step 4: fi nd $e_r = \underline{\min_l \|\Omega - \Omega^l\|}$

Step 5: if $e_r < T_r$, then Γ is recognized as face l from the training set.

So, the procedure for the face recognition is very similar to the face recognition by PCA. First, I have to do the normalization of the input data, that means from the original face the mean face is subtracted. After this, I have to determine the Fisherface, I have to determine. That means, I am considering the weights. The weights are w1, w2, w3, like this; that I have already explained in the PCA.

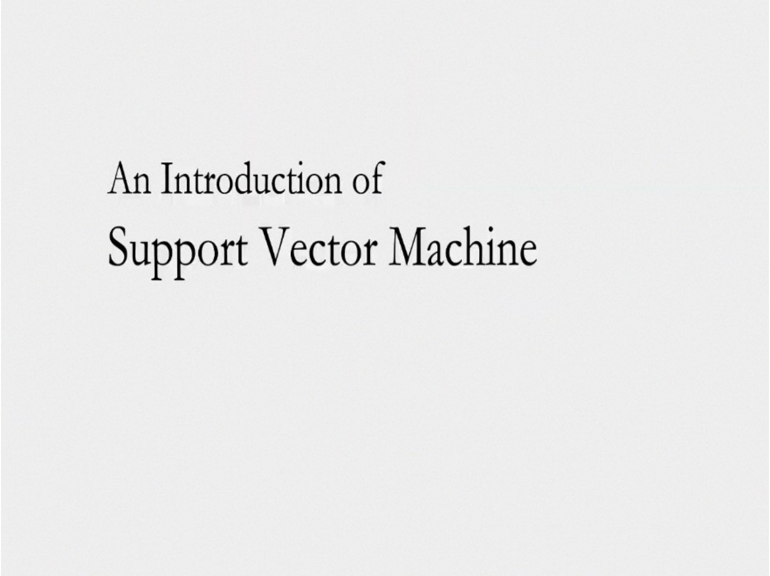
So, that means any unknown face can be represented by a linear combination of Fisherfaces. And suppose, a new face is coming. So, a new face can be also represented by a linear combination of Fisherfaces. After this, what I have to do for recognition? Just I have to compare the weights. So just, I have to compare the weights. I have to compare the weights. One is the weight corresponding to the training, and another one is the weights corresponding to the input test face. So, I have to compare the weights. And based on this comparison, if it is less than a particular threshold; so based on this condition I can recognize a particular face.

So, this concept is very much similar to the face recognition by PCA. But in this case, what I am considering? I am considering the Fisherface, I am considering. Corresponding to this, I am determining the transformation matrix. The transformation matrix is U, that is obtained from the eigenvectors. The eigenvectors of ${S_W}^{-1}$ into S_B.

In case of the PCA, we consider the transformation matrix A. The transformation matrix is obtained from the eigenvectors of the covariance matrix of the input data. So, C_X is the covariance matrix of the input data. And I am determining the eigenvectors. And from the eigenvectors, I can determine the transformation matrix.

In case of the LDA, what I am considering? I am again considering the eigenvectors of this, ${S_W}^{-1}$ into S_B. And from this, I can determine the transformation matrix, the transformation matrix is U. So, these 2 concepts are very similar. The face recognition by PCA and the face recognition by LDA. Now next, I will discuss the concept of the Support Vector Machine.

(Refer Slide Time: 42:17)

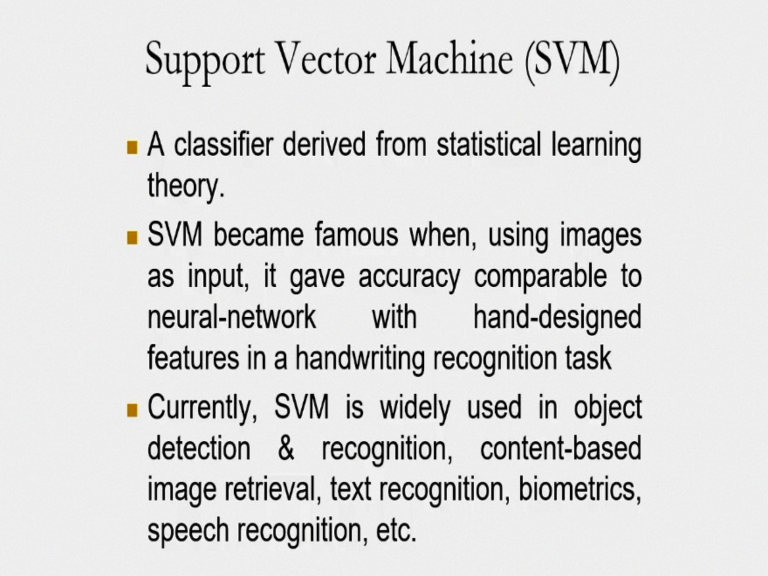


An Introduction of

Support Vector Machine

Now, I will discuss the concept of Support Vector Machine. An introduction of Support Vector Machine. So briefly, I will explain the concept of the Support Vector Machine. So, what is Support Vector Machine?

(Refer Slide Time: 42:31)



Support Vector Machine (SVM)

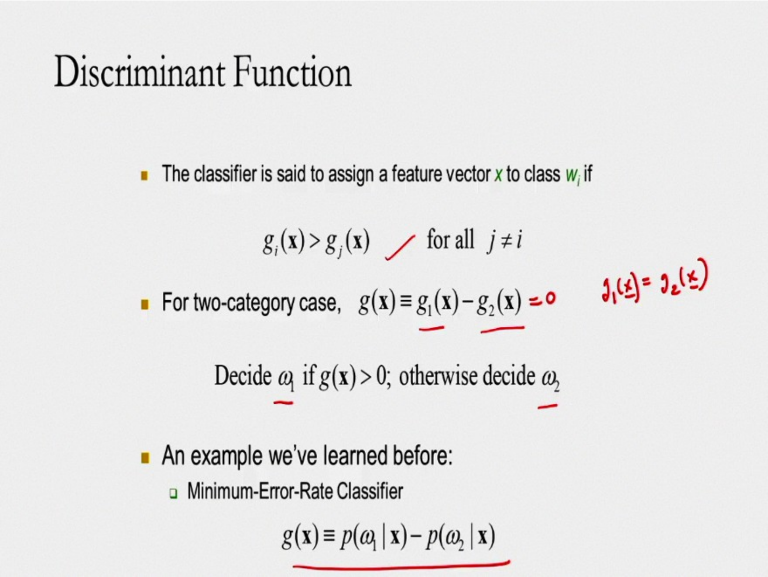
- A classifier derived from statistical learning theory.
- SVM became famous when, using images as input, it gave accuracy comparable to neural-network with hand-designed features in a handwriting recognition task
- Currently, SVM is widely used in object detection & recognition, content-based image retrieval, text recognition, biometrics, speech recognition, etc.

So, Support Vector Machine is a classifier derived from statistical learning theory. And already, I have explained, it is the discriminative model. Because in this case, I do not need the information of class conditional density. So, I have to determine the best decision boundary between 2 classes. So, if I consider more number of classes, I have to find the best decision boundaries between the classes. So, that is why, it is called a discriminative classifier; because, I do not need the information of that class conditional density.

And, Support Vector Machine, we can consider different applications; like handwriting character recognition, that is one applications. And there are many other applications, like object detection, and recognition, content-based image retrieval, text recognition, biometrics, speech recognition. So, there are many applications of Support Vector Machine, which can be used for classification and recognition.

So for this, we may consider hand-crafted features for classification. So, like this already I have explained some hand-crafted features; like color feature, texture features, or maybe the HOG, SIFT, I can consider. And based on these hand-crafted features, I can do the classification by Support Vector Machine.

(Refer Slide Time: 43:52)



Discriminant Function

- The classifier is said to assign a feature vector x to class w_i if

$$g_i(\mathbf{x}) > g_j(\mathbf{x}) \quad \text{for all } \; j \neq i$$

- For two-category case, $g(\mathbf{x}) \equiv \underline{g_1}(\mathbf{x}) - \underline{g_2(\mathbf{x})} = 0$ $\quad g_1(\underline{x}) = g_2(\underline{x})$

$$\text{Decide } \underline{\omega_1} \text{ if } g(\mathbf{x}) > 0\text{; otherwise decide } \underline{\omega_2}$$

- An example we've learned before:
  - Minimum-Error-Rate Classifier

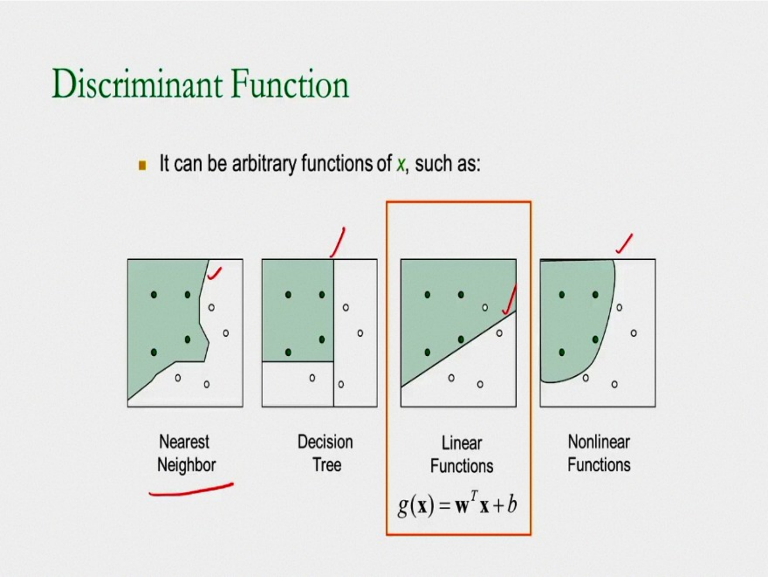
$$\underline{g(\mathbf{x}) \equiv p(\omega_1 \mid \mathbf{x}) - p(\omega_2 \mid \mathbf{x})}$$

So, you know this condition, that is the discriminate function, you know. So, already I have explained about the discriminate function. Now, this feature vector x can be assigned to a particular class, the particular class is w i. Based on this condition, that condition is, if g $_i$ (x) is greater than g $_j$ (x), and j is not equal to i. So, corresponding to this, I can assign a feature vector x to the class, the class is w $_i$. This is based on the discriminate function. And if I consider 2 classes, that is the 2-category case.

So, I can determine g_1x and also I can determine g_2x. So, g_1x minus g_2x, that is nothing but gx. Suppose g x is equal to 0, that corresponds to the decision boundary, that already I have explained. So, g x is equal to 0 means, it is the decision boundary. In the decision boundary, g_1x is equal to g_2x. So, x is the feature vector. So, based on g x, I can take a classification decision. So, I can consider or I can decide the class w1, if g x is greater than 0. Otherwise, I have to consider the class, the class is w2.

And, for the Minimum-Error-Rate Classifier, and that already I have explained. So, g x can be presented like this. So, g x is nothing but g_1x minus g_2x. So, what is g_1x? That is nothing but p the probability of w1 given x, that is posterior probability. And similarly, for g_2x the probability, the posterior probabilities, probability of w2 given x. So, for each and every class I have to determine g x, and I have to find a maximum discriminant function, and based on this I can take a classification decision.

(Refer Slide Time: 45:57)



Discriminant Function

- It can be arbitrary functions of *x*, such as:

Nearest Neighbor

Decision Tree

Linear Functions

$g(\mathbf{x}) = \mathbf{w}^T\mathbf{x} + b$

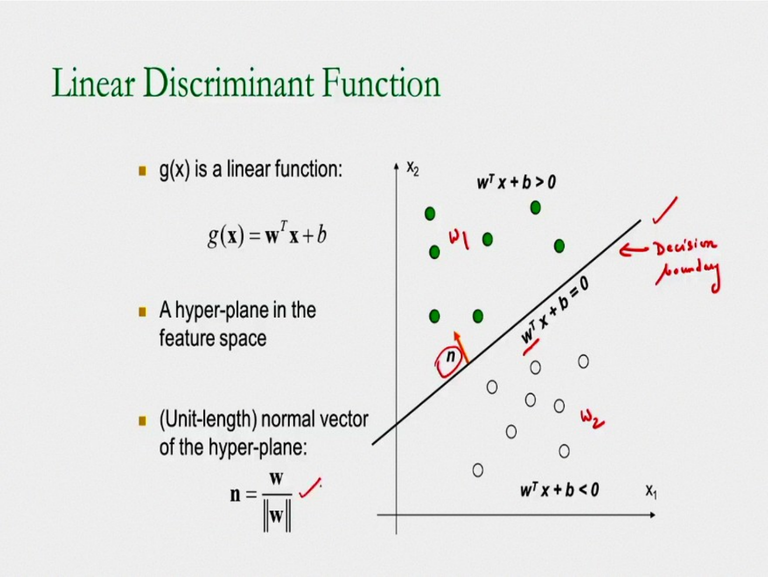
Nonlinear Functions

And here, I have shown some decision boundaries; you can see. First time, I am considering the nearest neighbor classification. So, you may get a decision boundary like this. So, this is the decision boundary between 2 classes. And in case of the decision tree, this is nothing but the binary decision, either yes or no. So, that type of decision I can consider by considering the decision tree, and corresponding to this I may get the decision boundary like this. That is the binary classification type, classifier.

And if I consider the g x is a linear function. So suppose, g x is equal to w ^{T} that is the transpose x plus b. So, w is the weight vector and x is the input feature vector plus b is the bias. So, corresponding to this, if I consider a linear function; then in this case, I will be getting a linear decision boundary, like this.

And also, I may get the nonlinear decision boundary between the classes. So, that last example is the nonlinear function, that is the nonlinear decision boundary, I can get. So, this is about that decision boundaries. So, I am now considering the discriminate function. The discriminate function is g x, that is equal to w ^{T} x plus b.

(Refer Slide Time: 47:18)



Linear Discriminant Function

- g(x) is a linear function:

$$g(\mathbf{x}) = \mathbf{w}^T\mathbf{x} + b$$

- A hyper-plane in the feature space
- (Unit-length) normal vector of the hyper-plane:

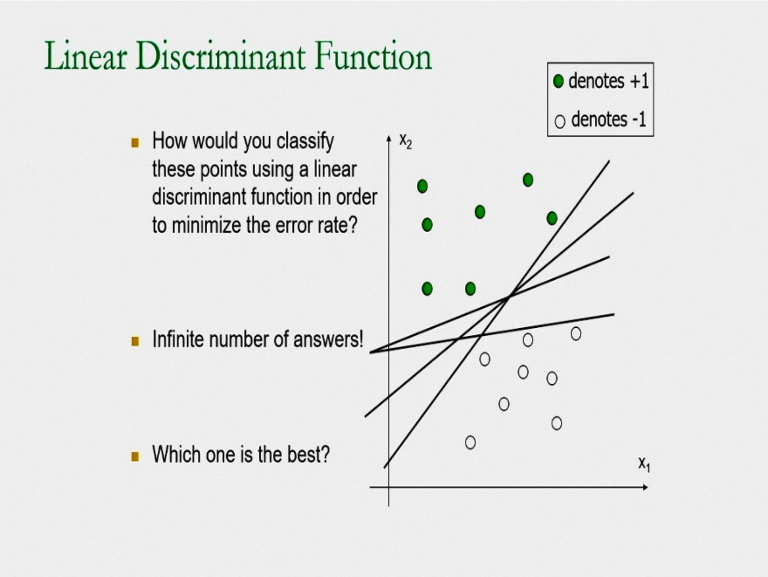
$$\mathbf{n} = \frac{\mathbf{w}}{\|\mathbf{w}\|} \checkmark$$

So, now g x is a linear function, that is the linear discriminant function, I am considering. So, g x is equal to w ^{T} x plus b. And I am considering a hyperplane, that is the decision boundary between 2 classes. You can see, here I am considering a 2-dimensional feature space, you can see x1 and x2 that is the 2-dimensional feature space, I am considering. And, you can see, this is the decision boundary. So, I am considering the decision boundary like this; decision boundary between the classes.

So, it is w ^{T} x plus b is equal to 0. So, that is the equation of the decision boundary. And suppose, w ^{T} x plus b is greater than 0, that corresponds to the class; suppose the class is w1, this class. And if w ^{T} x plus b less than 0, that we have considered the class, the class is w2. So, these 2 class, I can consider; one is the w1, another one is the w2. This w is different, this is the weight vector. So, w1 and w2, I am considering as classes, 2 classes I am considering.

And unit normal; that is the unit-length normal vector of the hyperplane, also I can determine. So. if you see this vector, this vector is the unit vector, that is the unit-length normal vector of the hyperplane, I can determine, that is nothing but w divided by the norm of w. So, that is unit vector, also you can determine.

(Refer Slide Time: 48:50)



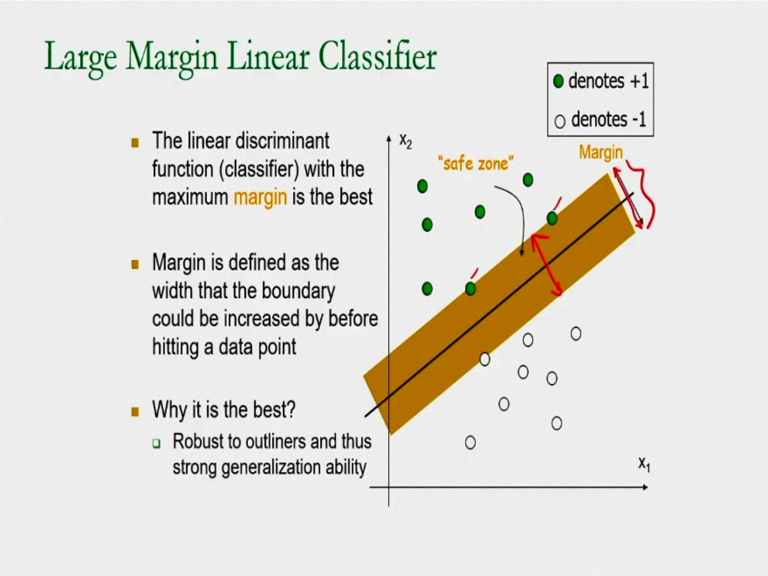
Linear Discriminant Function

- How would you classify these points using a linear discriminant function in order to minimize the error rate?
- Infinite number of answers!
- Which one is the best?

Now, how will you classify at these sample points using a linear discriminate function, in order to minimize the error rate? So, that is the concept. So, I am considering 2 classes. So, the first class is denoted by plus 1, and second class is denoted by minus 1. And, it is a 2-dimensional feature space, I am considering. So, infinite number of answers. Because I have to minimize the error rate. And you can see, I am showing the decision boundary between these 2 classes.

Again I may consider another decision boundary between these 2 classes; or maybe, I may consider another decision boundary between these 2 classes; or I may consider this decision boundary between these classes. So, I may get the number of decision boundaries. But which one is the best decision boundary; that I have to determine in the Support Vector Machine. So, which one is the best decision boundary between these 2 classes; that I have to determine by considering some optimization criterion, that I am going to explain in my next slide.

(Refer Slide Time: 49:56)



Large Margin Linear Classifier

- The linear discriminant function (classifier) with the maximum margin is the best
- Margin is defined as the width that the boundary could be increased by before hitting a data point
- Why it is the best?
  - Robust to outliners and thus strong generalization ability

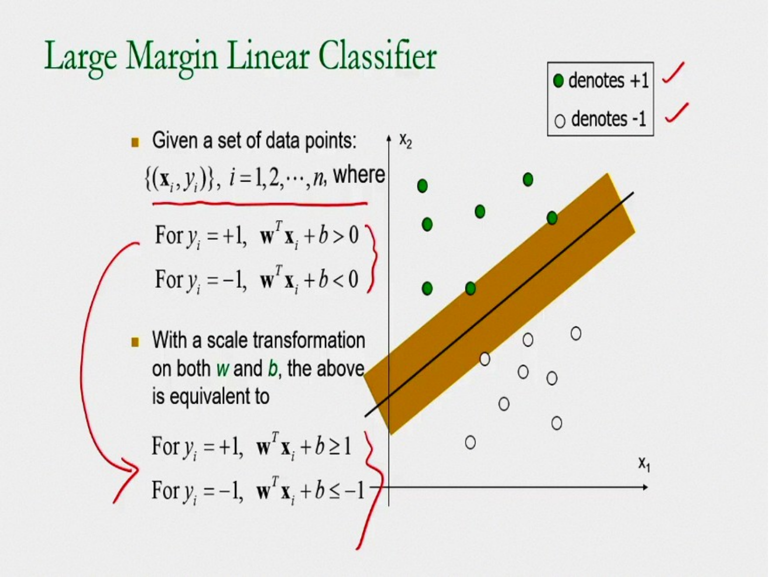
So, the linear discriminant function, or the classifier with the maximum margin is the best. So, this is the definition of the best decision boundary. So, what is the definition of the margin, that you can see here? So, margin is defined as the width, that the boundary could be increase by before hitting a data point. So, you can see, I am considering the boundary like this, and I am increasing the width of the boundary. And so that, it will just touch the sample points. That means the, you can see the vector these are the sample points. So, before hitting the data points, I can stop.

So based on this, I can define the margin. The margin of this, the hyperplane. So, this is the definition of the margin. So, beyond this, I cannot increase the width, because it will touch the data points. So, just before the hitting the data points, I have to stop. And corresponding to this, if I consider, this is the width of the decision boundary; suppose that corresponds to the margin. So, which is the best decision boundary, I want to determine?

Because in my previous slide, I have shown, I can draw number of decision boundaries between these 2 classes. But, which one is the best, that I can determine based on the margin. If the margin is more, then that will be good. That means, for a good decision boundary, the margin should be more, or the margin should be high. So, that based on this margin condition, I can find the best decision boundary between the classes.

The best decision boundary means, it is robust to outliers, and thus strong generalization ability. So here, you can see I am considering the safe zone. Because beyond this, I cannot increase the margin, because it will touch the data points. So, that corresponds to the safe zone. So, this is the safe zone, I am getting based on the margin.

(Refer Slide Time: 52:03)



Large Margin Linear Classifier

- Given a set of data points: $\{(\mathbf{x}_i, y_i)\},\ i = 1,2,\cdots,n$, where

  For $y_i = +1,\ \mathbf{w}^T\mathbf{x}_i + b > 0$

  For $y_i = -1,\ \mathbf{w}^T\mathbf{x}_i + b < 0$

- With a scale transformation on both *w* and *b*, the above is equivalent to

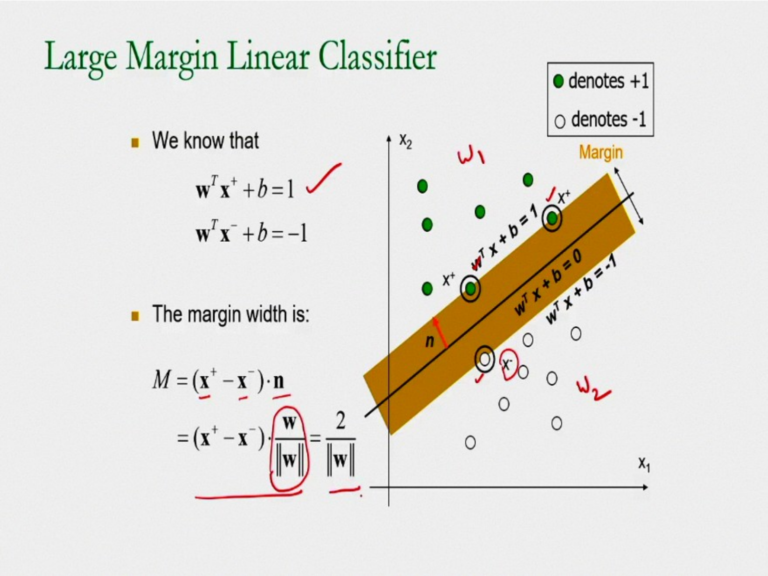
  For $y_i = +1,\ \mathbf{w}^T\mathbf{x}_i + b \ge 1$

  For $y_i = -1,\ \mathbf{w}^T\mathbf{x}_i + b \le -1$

So, how to determine the best decision boundary, that means how to maximize the margin between the classes. So, suppose the large margin linear classifier, that I want to design; that means, the margin should be the maximum. And already, I have mentioned. So, I have 2 classes; one is the plus, another one is, plus 1; another one is minus 1. So, given the data points, I have the data points; x_i, y_i, like this; i is equal to 1 to n. And for the class, that is y_i is equal to plus 1. So, corresponding to this, w ^{T} x_i plus b should be greater than 0. And similarly for the second class, y_i is equal to minus 1; w T xi plus b should be less 0. So, these are the conditions.

And after this, I can do some scale transformation, for both w and the b, and corresponding to this, these 2 equations will be equivalent to this. So, y_i is equal to plus 1, that corresponds to w ^{T} x_i plus b greater than equal to 1. And for y_i is equal to minus 1 w ^{T} x_i plus b should be less than equal to minus 1. So, I will be getting these 2 new conditions. So, these 2 new conditions, I am obtaining after the scale transformation on both w and b.

(Refer Slide Time: 53:34)



Large Margin Linear Classifier

- We know that

$$\mathbf{w}^T\mathbf{x}^+ + b = 1 \checkmark$$

$$\mathbf{w}^T\mathbf{x}^- + b = -1$$

- The margin width is:

$$M = (\mathbf{x}^+ - \mathbf{x}^-)\cdot \mathbf{n}$$

$$= (\mathbf{x}^+ - \mathbf{x}^-)\cdot \frac{\mathbf{w}}{\|\mathbf{w}\|} = \frac{2}{\|\mathbf{w}\|}$$

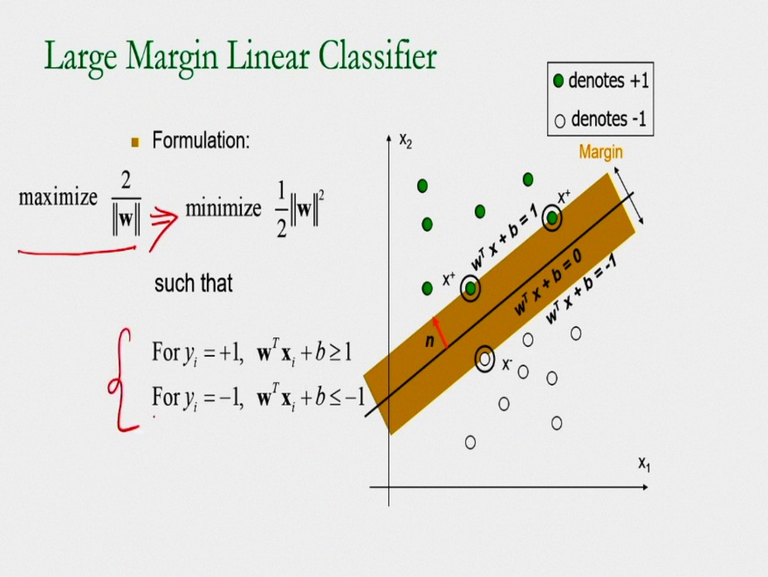
So you can see in this figure, I am considering the data points. So, if you see in the next figure here, you can see, I am considering x plus, x plus and x minus. So, this is the x plus, this is the x plus, and this is the x minus. That means, the margin is about to touch the data points. So, in one side, it is x plus; in another side, it is x minus. So, this x plus and the x minus, they are called the support vectors. Because this margin is about to touch the data points. Beyond this, I cannot increase the width of the margin.

And what is the objective? My objective is to get the large margin linear classifier. So, I have to get the maximum width of the margin. So, that is my objective; that is my goal. And based on this, I am defining the support vectors. The support vectors are x plus and the x minus. So, x plus, corresponding to the class w1, suppose. And another one is x minus, corresponding to the class w2. So, I have these support vectors. So, with the help of this support vectors, I can define the margin. So here, you can see, these are the support vectors, x plus and the x minus. These are the support vectors.

So, we know this condition, the w ^{T} x plus b is equal to 1, corresponding to the first class. So, I am now considering the support vector. So in this equation, w ^{T} x plus; x plus is the support vector, I am considering; plus b is equal to 1, that I am considering. And again, I am considering w ^{T} x minus. So, that x minus support vector, I am considering; plus b is equal to minus 1. So, these 2 equations, I am considering for 2 classes. The classes are w1 and w2.

And now, I want to determine the width of the margin. So, the margin width can be determined like this M is equal to x plus minus x minus. So, this is one support vector, and x minus is another support vector. And I am considering the normal to the hyperplane. So, you get the, this unit normal to the hyperplane, I am considering. So, this unit normal already I have determined. So, this is the unit normal, that is the n. And this is nothing but it is equal to 2 divided by w norm. So, I am just determining the norm of w. w is the weight vector.

(Refer Slide Time: 56:08)



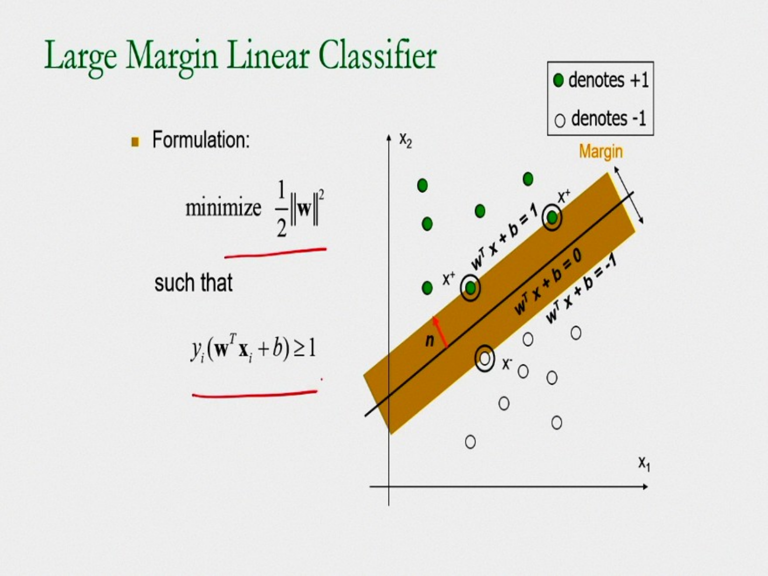
Large Margin Linear Classifier

- Formulation:

$$\text{maximize } \frac{2}{\|\mathbf{w}\|} \Rightarrow \text{minimize } \frac{1}{2}\|\mathbf{w}\|^2$$

such that

$$\text{For } y_i = +1, \ \mathbf{w}^T\mathbf{x}_i + b \geq 1$$
$$\text{For } y_i = -1, \ \mathbf{w}^T\mathbf{x}_i + b \leq -1$$



Large Margin Linear Classifier

- Formulation:

$$\text{minimize } \frac{1}{2}\|\mathbf{w}\|^2$$

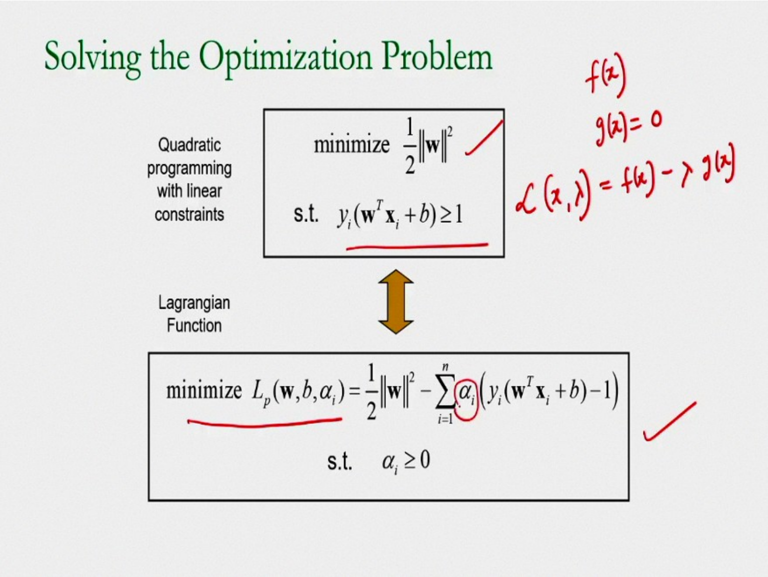
such that

$$y_i(\mathbf{w}^T\mathbf{x}_i + b) \geq 1$$

Now, for the large margin linear classifier, what I am considering? I am considering the formulation, that I have to maximize the margin. So, the width of the margin is 2 divided by w norm. So, I have to maximize this. And, that is the goal of a large margin linear classifier in the Support Vector Machine. And based on this, what are the conditions? The condition is y $_i$ is equal to plus 1. For this condition, the condition is w ^{T} x $_i$ plus b greater than equal to 1. And what is another class? Another class is y $_i$ is equal to minus 1. That is another class; and this condition is w ^{T} x $_i$ plus b less than equal to minus 1.

And again the formulation, I have to maximize 2 divided by w norm; that is equivalent to minimizing 1 divided 2 w norm whole square. So, this is actually equivalent to this. The maximizing 2 divided by w norm is equivalent to minimizing 1 by 2 w norm whole square. Such that these 2 conditions, I have to consider. So, formulation is this. So, minimize 1 by 2 w norm whole square, that I have to minimize and condition is this.

(Refer Slide Time: 57:31)



Solving the Optimization Problem

Quadratic programming with linear constraints

$$\text{minimize } \frac{1}{2}\|\mathbf{w}\|^2 \quad ✓$$

$$\text{s.t. } \underline{y_i(\mathbf{w}^T\mathbf{x}_i + b) \geq 1}$$

Lagrangian Function

$$\underline{\text{minimize } L_p(\mathbf{w}, b, \alpha_i)} = \frac{1}{2}\|\mathbf{w}\|^2 - \sum_{i=1}^{n} \alpha_i \left(y_i(\mathbf{w}^T\mathbf{x}_i + b) - 1 \right) \quad ✓$$

$$\text{s.t. } \alpha_i \geq 0$$

$f(x)$

$g(x) = 0$

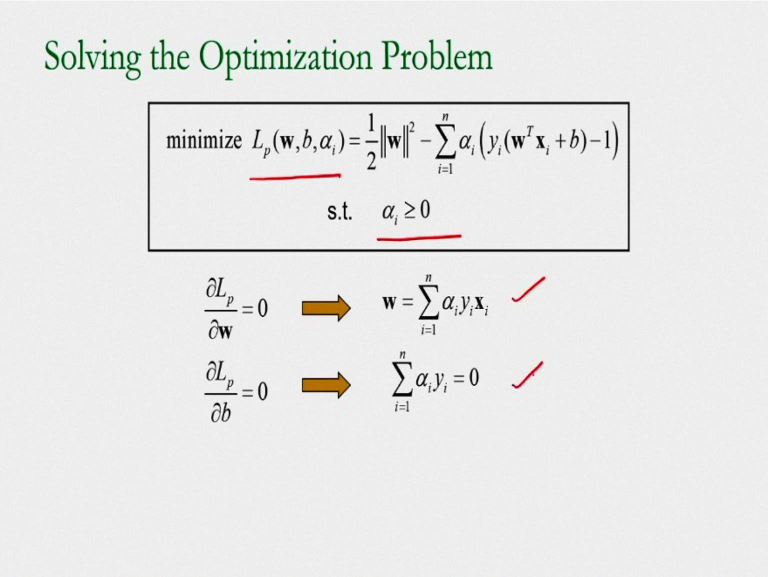
$\mathcal{L}(x, \lambda) = f(x) - \lambda\, g(x)$

So, here I am considering this optimization problem. So, I have to minimize these, subject to the condition; the condition is this. So, this a optimization problem. So, how to solve this optimization problem? So, for solving the optimization problem, I can consider Lagrangian function. So, in mathematical optimization, Lagrange's multiplier method is used, you know that condition. So, you have to see the mathematics book. In mathematical optimization, Lagrange's multiplier method is used. And it is used to find local maxima and the minima of a function, subject to some constraints. That means, subject to the condition that one or more equation have to be satisfied exactly by the chosen value of the variables.

So, I am repeating this. It is used to find the local maxima and the minima of a function, subject to some constraints. That means, subject to the conditions, that one or more equations have to be satisfied exactly by the chosen value of the variables. In order to find the maxima, or the minima of the function. Suppose the function is f x. So, I want to find the maxima or the minima of the function f x, subject to the equating constraint. So, constraint is suppose g x is equal to 0. So, this constraint I am considering.

Then this Lagrangian function, I can write like this. This is a Lagrangian function, x, λ. So, lambda is the Lagrange's multiplier, is equal to $f x - \lambda g x$. So, I can write like this. So, for more details you have to see the mathematics books. So, how to do the optimization by considering the Lagrange's function. So, this is the Lagrange's function, and I have to minimize this one. So, α_i is the Lagrange's multiplier. So, this is the Lagrange's multiplier.

(Refer Slide Time: 59:38)



Solving the Optimization Problem

$$\text{minimize } \underline{L_p(\mathbf{w},b,\alpha_i)} = \frac{1}{2}\|\mathbf{w}\|^2 - \sum_{i=1}^{n}\alpha_i\left(y_i(\mathbf{w}^T\mathbf{x}_i + b) - 1\right)$$

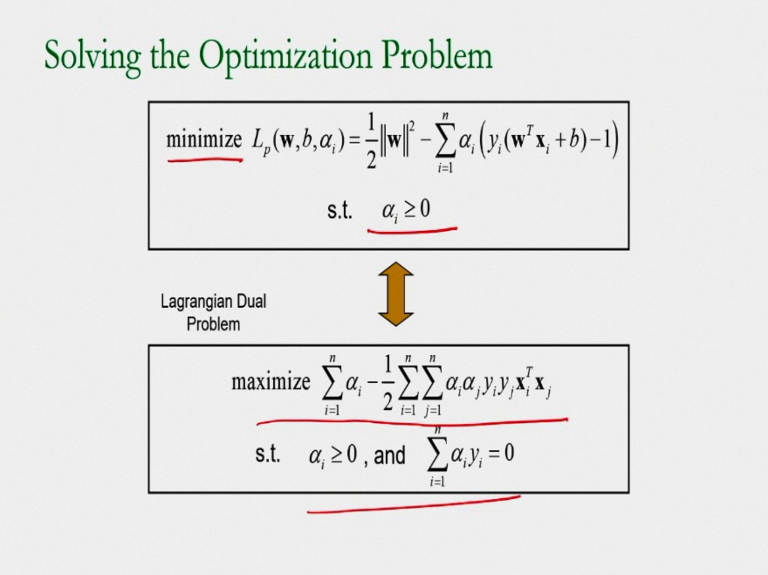
$$\text{s.t. } \underline{\alpha_i \ge 0}$$

$$\frac{\partial L_p}{\partial \mathbf{w}} = 0 \Rightarrow \mathbf{w} = \sum_{i=1}^{n}\alpha_i y_i \mathbf{x}_i \quad ✓$$

$$\frac{\partial L_p}{\partial b} = 0 \Rightarrow \sum_{i=1}^{n}\alpha_i y_i = 0 \quad ✓$$

So again, I am writing this. So, I have to minimize this function, the Lagrangian function and subject to the condition, the condition is, α_i should be greater than equal to 0. So, α_i is the Lagrange’s multiplier. And for this, I am just taking the derivative of L p with respect to the weight vector, the weight vector is w. And similarly, the del L p divided by del b should be equal to 0. So, based on these differentiations, I want to find the maximum or the minimum conditions. So, based on this the partial derivative, I want to find the value of w and this.

(Refer Slide Time: 60:23)



Solving the Optimization Problem

$$\text{minimize } L_p(\mathbf{w}, b, \alpha_i) = \frac{1}{2}\|\mathbf{w}\|^2 - \sum_{i=1}^{n} \alpha_i \left(y_i(\mathbf{w}^T\mathbf{x}_i + b) - 1 \right)$$

$$\text{s.t.} \quad \alpha_i \geq 0$$

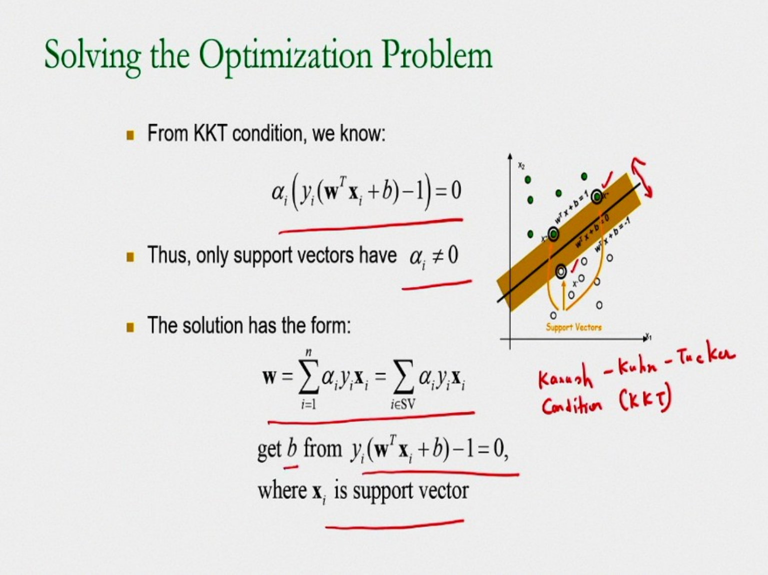
Lagrangian Dual Problem

$$\text{maximize } \sum_{i=1}^{n} \alpha_i - \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j \mathbf{x}_i^T \mathbf{x}_j$$

$$\text{s.t.} \quad \alpha_i \geq 0 \text{ , and } \sum_{i=1}^{n} \alpha_i y_i = 0$$

And this is nothing but the Lagrangian Dual Problem. That means you can see, because my objective is to minimize this function the Lagrangian function, subject to condition. This condition is important, that is equivalent to maximizing this. That is equivalent to maximizing this subject to this, these 2 conditions. And this is called the Lagrangian Dual Problem. So, you see that mathematics. Again, I am repeating this, you see the mathematics, how to solve the optimization problem using the Lagrange's function.

(Refer Slide Time: 60:54)



Solving the Optimization Problem

- From KKT condition, we know:

$$\alpha_i \left(y_i(\mathbf{w}^T\mathbf{x}_i + b) - 1 \right) = 0$$

- Thus, only support vectors have $\alpha_i \neq 0$

- The solution has the form:

$$\mathbf{w} = \sum_{i=1}^{n} \alpha_i y_i \mathbf{x}_i = \sum_{i \in SV} \alpha_i y_i \mathbf{x}_i$$

get b from $y_i(\mathbf{w}^T\mathbf{x}_i + b) - 1 = 0$,

where $\mathbf{x}_i$ is support vector

Karush - Kuhn - Tucker Condition (KKT)

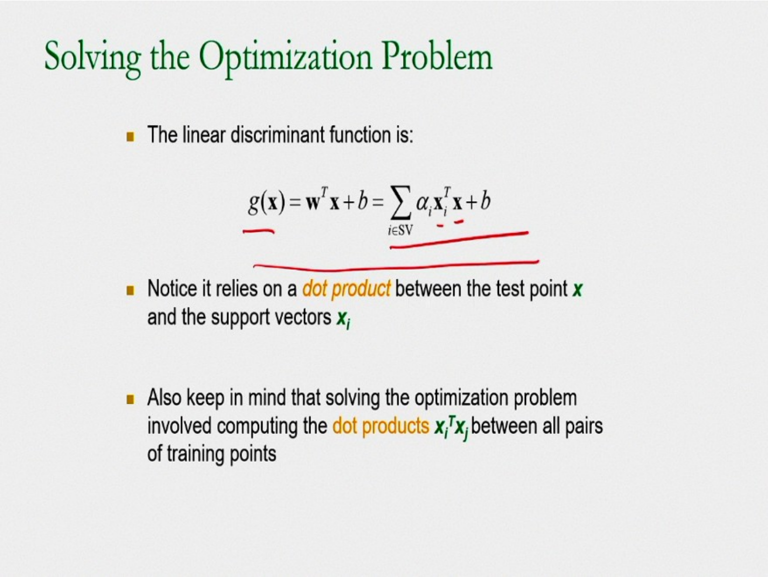
And now I am considering, the KKT condition. What is the KKT condition? So, I am considering. So, from the KKT condition, the KKT condition is nothing but, the KKT means Karush-Kuhn-Tucker condition, that also you have to see. This is called the KKT. It is used in mathematical optimization. It is nothing but the first derivative test. It is also called the first order necessary condition for a solution in a nonlinear programming to be optimal, provided that some regularity conditions are satisfied.

So, this is the briefly, what is the KKT. So, it is used in mathematical optimization and it is nothing but the first derivative test. And it is also called the first order necessary conditions for a solution in a nonlinear programming to be optimal, provided that some regularity conditions should be satisfied. So, by considering this KKT, I am considering this one, that is α_i into y_i and after this, I am considering $w^T x_i$ plus b minus 1 is equal to 0.

That I am considering; and thus, only for the support vector that is $\alpha_i$is not equal to 0, that is only support vector have this condition. That is $\alpha_i$is not equal to 0. So, the solution will be like this. So, I will be getting the solution of this. So, I will be getting the value of w and also, I have to consider this one to get b from this. So, I can get b from this equation. And in this case xi is the support vector. So, I am considering xi is the support vector.

And in the figure also, I have shown the support vectors, and you can see this is the margin I am considering. So, x plus and the x minus, these are the support vectors. And based on this, you can see, based on this Lagrangian's multiplier function, Lagrange's function, I am getting the solution. And you can apply the KKT, this condition, and I will be getting the value of w, and also, I can get the value of b, the b also I can determine from this.

(Refer Slide Time: 63:27)



Solving the Optimization Problem

- The linear discriminant function is:

$$\underline{g}(\mathbf{x}) = \mathbf{w}^T\mathbf{x} + b = \underline{\sum_{i\in \mathrm{SV}} \alpha_i \mathbf{x}_i^T \mathbf{x} + b}$$

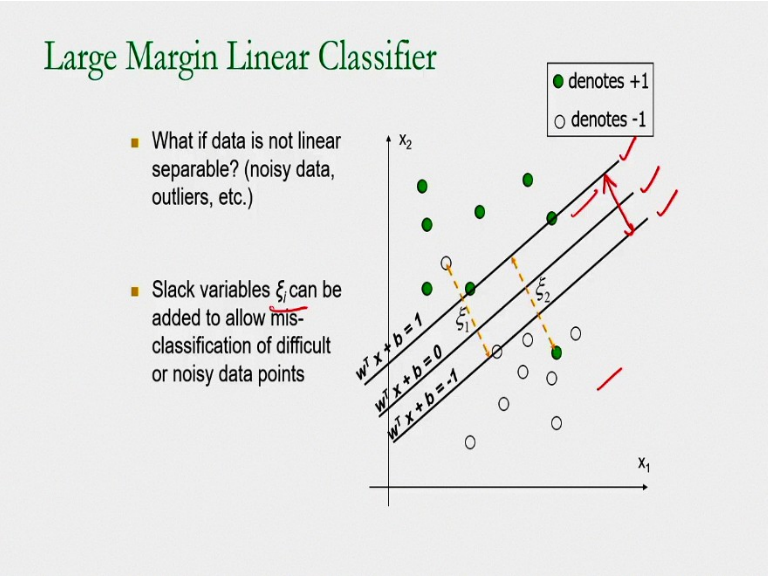
- Notice it relies on a *dot product* between the test point ***x*** and the support vectors $\boldsymbol{x}_i$

- Also keep in mind that solving the optimization problem involved computing the dot products $\boldsymbol{x}_i^T\boldsymbol{x}_j$ between all pairs of training points

And finally, the linear discriminant function I will be getting like this, w ^{T} into x plus b is nothing but, I will be getting this one. And only I am considering the support vectors, i is mainly the support vector, I am considering. So, alpha i is nothing but, it is the Lagrange's multiplier. So, i is nothing but the support vector. So, you can see in this expression, in the g x expression, that is the discriminate function, the linear discriminate function. So, it is nothing but the dot product between the test point. So, test point is x, and the support vectors are xi. So, first I have to determine the dot product between the test point x and the support vectors xi, and based on this I can determine g x.

So, suppose the new test vector is coming, suppose the new test vector is coming. So, for this one I have to do, I have to find out the dot product between the test point x and the support vector xi. And from this, I can determine the discriminate function, the discriminate function is g x. Also, keep in mind that solving the optimization problem involve computing the dot product between x $_i$ ^{T} and x j between all pairs of the training points. So, for all pairs of the training points I have to find the dot products, that is the condition. So, based on this, this discriminate function I can do the classification by considering that support vectors.

(Refer Slide Time: 64:59)



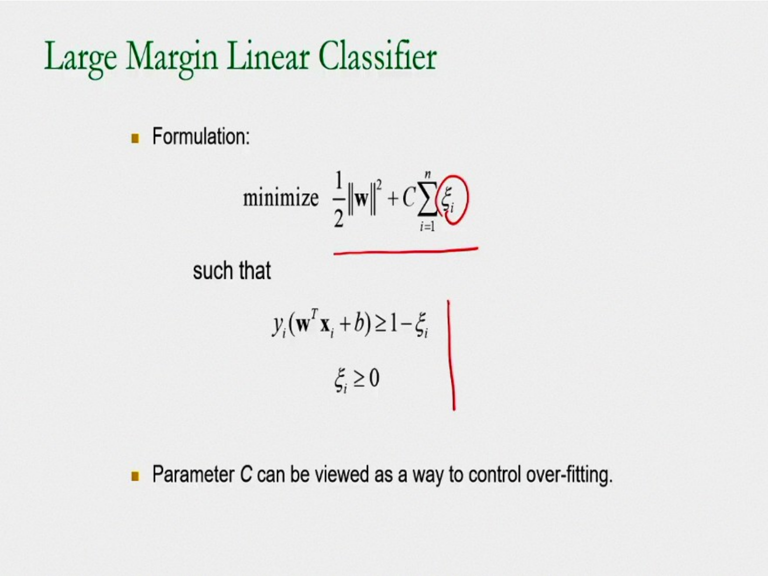
Large Margin Linear Classifier

- What if data is not linear separable? (noisy data, outliers, etc.)
- Slack variables ξ_i can be added to allow mis-classification of difficult or noisy data points

And suppose if I consider, if the data is not linear, linearly separable. So, suppose if I consider the noisy data points or the outliers. And again here, I am showing 2 classes, one is the plus green, and another one is the white that is the minus, 2 classes I am considering. And I am considering the noisy data points and outliers. So, for this what is the formulation? The formulation is I have to consider slack variables. So, slack variable I am considering. So, it can be added to allow misclassification of difficult or the noisy data points. These slack variables are introduced to allow certain constraints to be violated.

So, slack variables are defined to transform an inequality expression into an equality expression, that is the mathematics, you can see. That is the slack variables, we can define to transform an inequality expression into an equality expression. So, that is the objective of the slack variables. And you can see, I am showing this equations w ^{T} x plus b is equal to 1 corresponding to this, this is the line; w ^{T} x plus b is equal to 0, that is the decision boundary; w ^{T} x plus b is equal to minus 1, that is the line. And you can see the margin here. I am showing the margin here.

(Refer Slide Time: 66:25)



Large Margin Linear Classifier

- Formulation:

$$\text{minimize } \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{n}\xi_i$$

such that

$$y_i(\mathbf{w}^T\mathbf{x}_i + b) \ge 1 - \xi_i$$

$$\xi_i \ge 0$$

- Parameter *C* can be viewed as a way to control over-fitting.

And corresponding to this slack variable problem, the formulation will be something like this, 1 by 2 w norm whole square and after this, plus C. So, C parameter I am considering and this is the slack variable. So, ξ_i is the slack variable. So, the conditions are like this. So, I have to consider these 2 conditions, and this parameter C can be viewed as a way to control overfitting. That means, the parameter C tells the optimization, how much you want to avoid misclassification by, is training examples. So, that means I have to avoid the misclassification and this parameter C controls the overfitting. So, that information I am giving with the help of the parameter, that parameter is C. So, this is the concept of the large margin linear classifier.

So, briefly I have explained the large margin linear classifier, and also their nonlinear Support Vector Machines. Suppose, if I consider many noisy points or the outliers. So, for this I can consider nonlinear Support Vector Machine and, in this case, I can consider the projection of the low dimensional data into high dimensional space. That means, I can do the projection of the low dimensional space into high dimensional space, and I can consider the nonlinear Support Vector Machine. I am not explaining the concept of the nonlinear Support Vector Machine. Briefly, I have explained the concept of the Support Vector Machine, that is the large margin linear classifier.

In this class, I discussed the basic concept of the LDA, and the Support Vector Machine. In case of the LDA, I have to find the best direction of the projection. So, which one is the best projection direction I have to find? And for this I have considered 1 criterion function, and for this I considered between-class scatter matrix and the within-class scatter matrix. So, based on this, I can find the best prediction direction; I can find.

In case of the Support Vector Machine, I determine the best decision boundary between 2 classes. So, that means I have to maximize the margin and between the 2 classes. And in this case based on this, the margin, the width of the margin, I can define the support vectors. And based on these support vectors, I can determine the discriminate function g x; and after this, I can do the classification.

The Support Vector Machine is a discriminative classifier, because I do not need the information of class conditional density. So, these concepts, 2 concepts; one is the LDA, another one is the Support Vector Machine, briefly I have explained in this class. So, let me stop here today. Thank you.