Pattern Recognition

Lecture 15. Linear Discriminant Functions: Minimum Squared Error Procedures and Fisher's Linear Discriminant

Dr. Shanshan ZHAO

shanshan.zhao@xjtlu.edu.cn

School of AI and Advanced Computing

Xi'an Jiaotong-Liverpool University

Academic Year 2021-2022

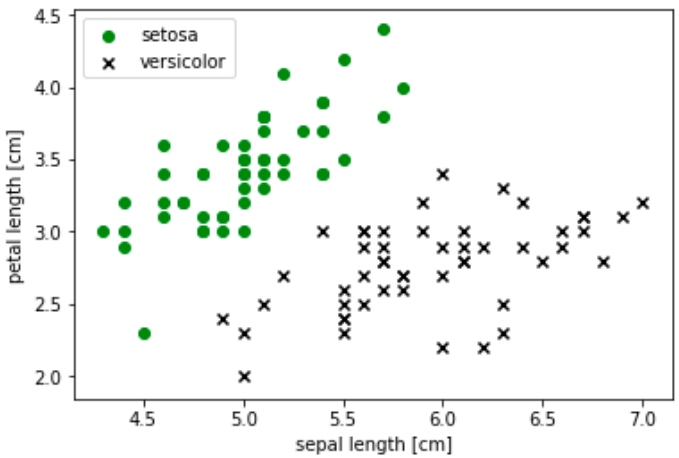
Table of Contents

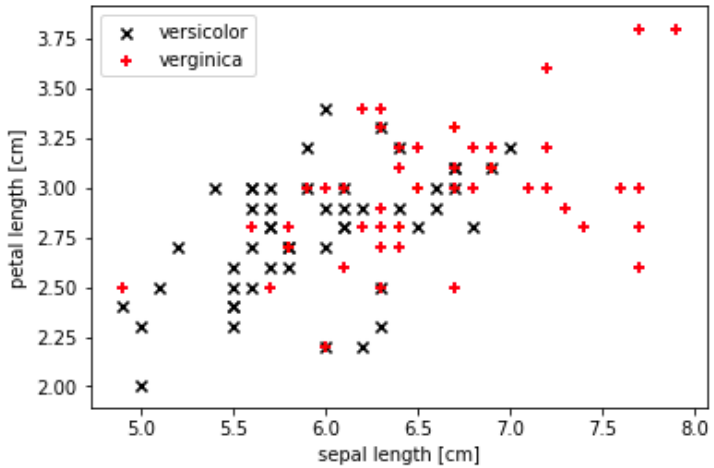
Recap

- **Classification based on Bayes Decision Theory**
  - Bayes Decision Theory
  - Minimizing the classification error probability
  - Minimizing the Average Risk
  - Discriminant functions for Normal densities and decision surfaces
    - reach a linear discriminant function when all classes share same covariance matrix
- **Parametric Estimation (parameters estimation for known probability)**
  - MLE
  - MAP
- **Non-Parametric Estimation (parameters esitmation for unknown probability)**
  - Kernel density estimation
  - KNN (from the density to classification)
- **Linear Discriminant functions (Linear classifiers: no assumption on the densities anymore)**
  - Geometry interpretation (Augmentation and Normalization)
  - Criterion function : Perceptron Criterion Function
  - Gradient descent

Perceptron Criterion Function

- If classes are linearly separable, the perceptron rule is guaranteed to converge to a valid solution
- However, if the two classes are not linearly separable, the perceptron rule will not converge.
  - Since there is no weight vector ***a*** can correctly classify every sample in a non-separable dataset, the corrections in the perceptron rule will never cease.





(a) Linearly Separable Data (IRIS Dataset) (b) Linearly Non-Separable Data (IRIS Dataset)

you might wanna check https://vitalflux.com/how-know-data-linear-non-linear/

Minimum Squared Error Procedures

The classical MSE criterion provides an alternative to the perceptron rule

Perceptron

- 1. Focused on the misclassified samples
- 2. seek a vector **a** making all inner product $a^T y_i$ positive
- 3. Try to find the solution to a set of linear inequalities

MSE

- 1. Involves all the samples
- 2. Seek a vector **a** making $a^T y_i = b_i$, where b_i is some arbitrarily specified positive constants
- 3. Find the solution to a set of linear quations

Minimum Squared Error Procedures

- The treatment of simultaneous linear equations is simplified by introducing matrix notation.
- Let Y be the n-by-$\hat{d}$ matrix ($\hat{d} = d + 1$) whose ith row is the vector y_i^T
- let b be the column vector $b = (b_1, ..., b_n)^T$
- Then our problem is to find a weight vector a satisfying

$$\begin{bmatrix} Y_{10} & Y_{11} & \cdots & Y_{1d} \\ Y_{20} & Y_{21} & \cdots & Y_{2d} \\ \vdots & \vdots & & \vdots \\ Y_{n0} & Y_{n1} & \cdots & Y_{nd} \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ \vdots \\ a_d \end{bmatrix} = \begin{bmatrix} b_0 \\ b_1 \\ \vdots \\ b_d \end{bmatrix}$$

or

$$Ya = b$$

Minimum Squared Error Procedures

- If Y were nonsingular, we could write $a = Y^{-1}b$ and obtain a formal solution at once.
- However, Y is rectangular, usually with more rows than columns. When there are more equations than unknowns, a is overdetermined, and ordinarily no exact solution exists.
- However, we can seek a weight vector a that minimizes some function of the error between Ya and b.
- If we define the error vector e by

$$e = Ya - b$$

- Then one approach is to try to minimize the squared length of the error vector. This is equivalent to minimizing the sum-of-squared-error criterion function

$$J_s(a) = ||Ya - b||^2 = \sum_{i=1}^{n}(a^T y_i - b_i)^2$$

Minimum Squared Error Procedures

The problem of minimizing the sum of squared error is a classical one. It can be solved by a gradient search procedure. A simple closed-form solution can also be found by forming the gradient

$$\nabla J_s = \sum_{i=1}^{n} 2(a^T y_i - b_i) y_i = 2Y^T(Ya - b)$$

and setting it equal to zero. This yields the necessary condition

$$Y^T Ya = Y^T b$$

If matrix $Y^T Y$ is square andnonsingular, we can solve a uniquely

$$a = (Y^T Y)^{-1} Y^T b$$

The MSE solution depends on the margin vector b, and we shall see that different choices for b give the solution different properties. If b is fixed arbitrarily, there is no reason to believe that the MSE solution yields a separating vector in the linearly separable case. However,it is reasonable to hope that by minimizing the squared-error criterion function we might obtain a useful discriminant function in both the separable and the nonseparable cases[1].

MSE example

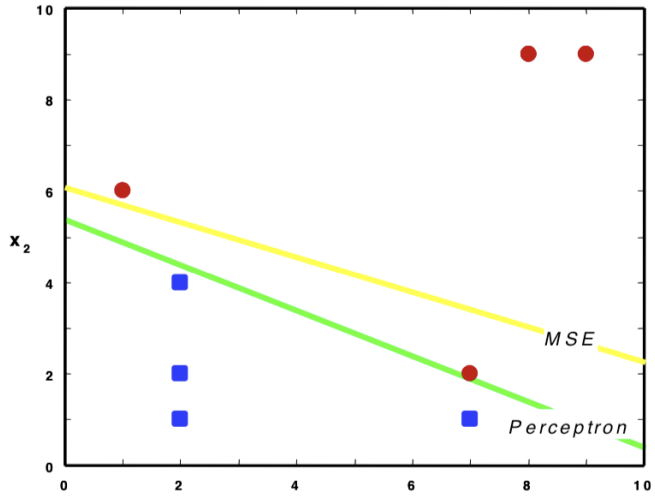
Compute the perceptron and MSE solution for the dataset

- $X1 = [(1,6), (7,2), (8,9), (9,9)]$
- $X2 = [(2,1), (2,2), (2,4), (7,1)]$

Perceptron leaning

- Assume $\eta = 0.1$ and an online update rule
- Assume $a(0) = [0.1, 0.1, 0.1]$
- SOLUTION
  - Normalize the dataset

$$Y = \begin{bmatrix} 1 & 1 & 6 \\ 1 & 7 & 2 \\ 1 & 8 & 9 \\ 1 & 9 & 9 \\ -1 & -2 & -1 \\ -1 & -2 & -2 \\ -1 & -2 & -4 \\ -1 & -7 & -1 \end{bmatrix}$$



  - Iterate through all the examples and update $a(k)$ on the ones that are misclassified
    - Y(1) $\Rightarrow$ [1 1 6]*[0.1 0.1 0.1]^T>0 $\Rightarrow$ no update
    - Y(2) $\Rightarrow$ [1 7 2]*[0.1 0.1 0.1]^T>0 $\Rightarrow$ no update

      ...
    - Y(5) $\Rightarrow$ [-1 -2 -1]*[0.1 0.1 0.1]^T<0 $\Rightarrow$ update a(1) = [0.1 0.1 0.1] + η[-1 -2 -1] = [0 -0.1 0]
    - Y(6) $\Rightarrow$ [-1 -2 -2]*[0 -0.1 0]^T>0 $\Rightarrow$ no update

      ....
    - Y(1) $\Rightarrow$ [1 1 6]*[0 -0.1 0]^T<0 $\Rightarrow$ update a(2) = [0 -0.1 0] + η[1 1 6] = [0.1 0 0.6]
    - Y(2) $\Rightarrow$ [1 7 2]*[0.1 0 0.6]^T>0 $\Rightarrow$ no update

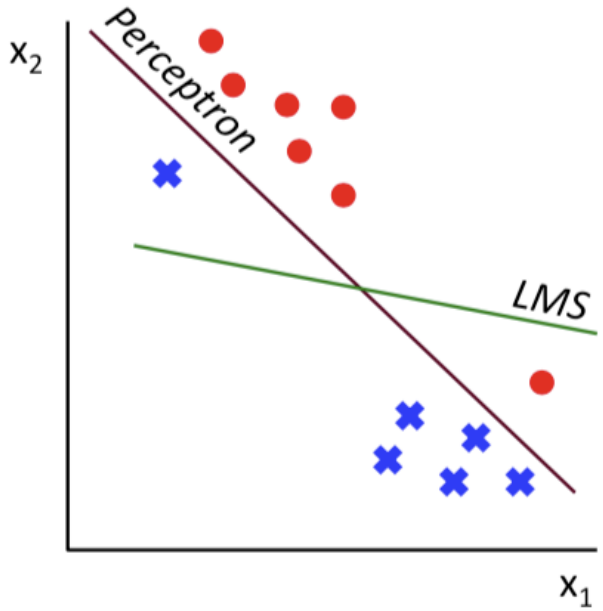
      ...
  - In this example, the perceptron rule converges after 175 iterations to $a = [-3.5\ 0.3\ 0.7]$
  - To convince yourself this is a solution, compute Ya (you will find out that all terms are non-negative)

MSE

- The MSE solution is found in one shot as $a = (Y^TY)^{-1}Y^Tb = [-1.1870\ \ 0.0746\ 0.1959]$
  - For the choice of targets b = [1 1 1 1 1 1 1]^T
  - As you can see in the figure, the MSE solution misclassifies one of the samples

Summary: perceptron vs. MSE

- **Percepton criterion**
  - The perceptron rule always find a solution if the classes are linearly separable, but does not converge if the classes are non-separable.
- **MSE criterion**
  - The MSE solution has guaranteed convergence, but it may not find a separating hyperplane if classes are linearly separable.
    - Notice that MSE tries to minimize the sum of the squares of distances of the training data to the separating hyperplane, as opposed to finding this hyperplane.



Relation to Fisher's Linear Discriminant

- We shall show that with the proper choice of the vector b, the MSE discriminant function at y is directly related to Fisher's linear discriminant.
- Assume that we have a set of n d-dimensional samples $x_1, ..., x_n$, n_1 of which are labelled ω_1, and n_2 of which are labelled ω_2.

$$\begin{bmatrix} I_1 & X_1 \\ -I_2 & X_2 \end{bmatrix}$$

Where I_i is a column vector of n_i ones, and X_i is an n_i-by-d matrix whose rows are samples lablled ω_i, we have

$$a = \begin{bmatrix} w_0 \\ w \end{bmatrix} b = \begin{bmatrix} \frac{n}{n_1} I_1 \\ \frac{n}{n_2} I_2 \end{bmatrix}$$

The special choice of b links to the MSE solution to **Fisher's Linear Discriminant**.

Fisher's Linear Discriminant

- to find a linear combination of features which characterizes or separates two or more classes of objects or events
- used as a linear classifier or dimensionality reduction

Fisher Linear Discriminant project to a line which preserves direction useful for data classification.

The transformation is based on maximizing the ratio of **"between-class variance"** to **"within-class variance"**, aiming at reducing data variation in the same class and in increasing the separation between class.

Fisher's Linear Discriminant

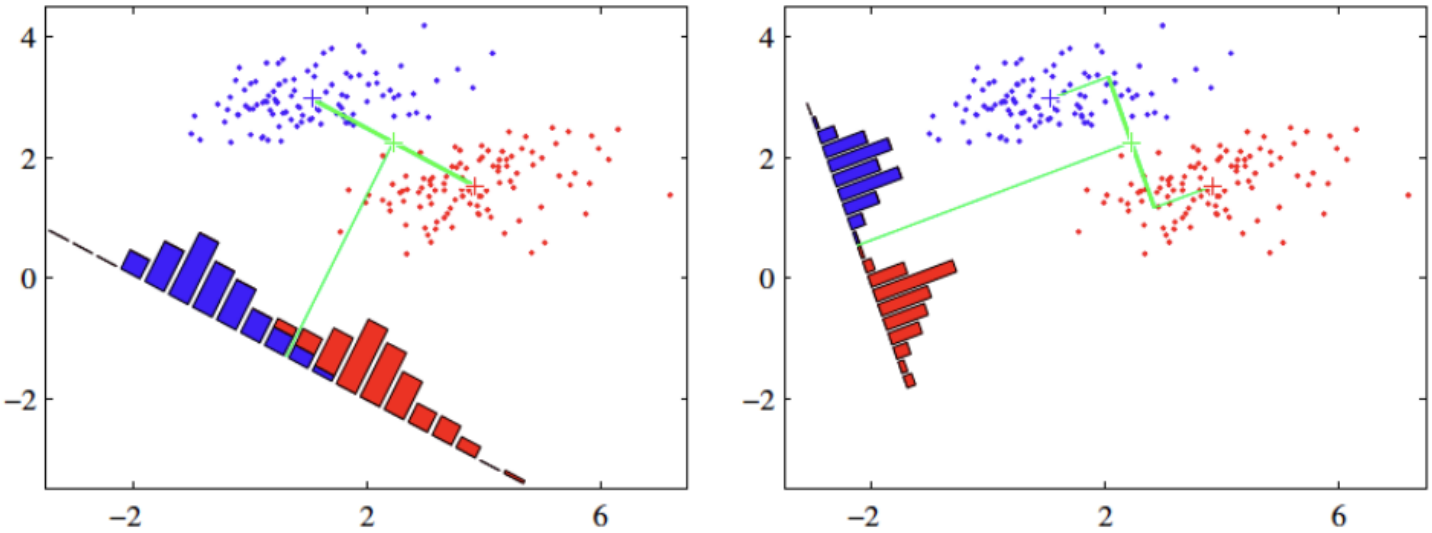
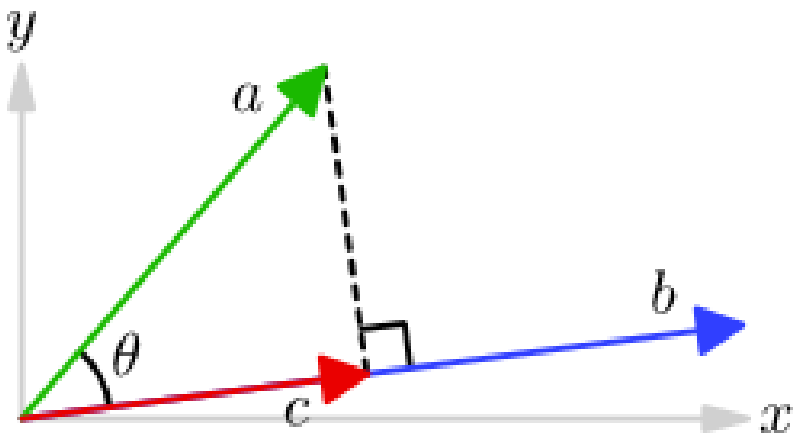


Figure: The left plot shows samples from two classes (depicted in red and blue) along with the histograms resulting from projection onto the line joining the class means. Note that there is considerable class overlap in the projected space. The right plot shows the corresponding projection based on the Fisher linear discriminant, showing the greatly improved class separation[2][3].

Fisher's Linear Discriminant

Preliminary 1. Projection from vector a to vector b

- Scalar $|c|$ is the projection from a to b.
- If b is a unit vector which means $|b| = 1$, then the scalar projection $|c|$ can be represented as $a \cdot b$.



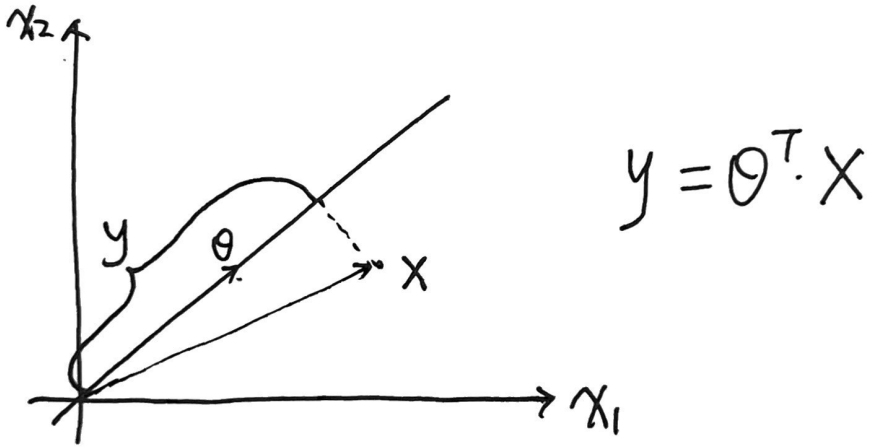
$$a.b = |a||b|cos(\theta)$$

$$a.b = a_x b_x + a_y b_y$$

$$|c| = |a|cos(\theta)$$

Fisher's Linear Discriminant

Our goal is seeking to obtain a **scalar** y by projecting the samples X onto a line:



Then try to find the θ^* to maximize the ratio of "between-class variance" to "within-class variance".

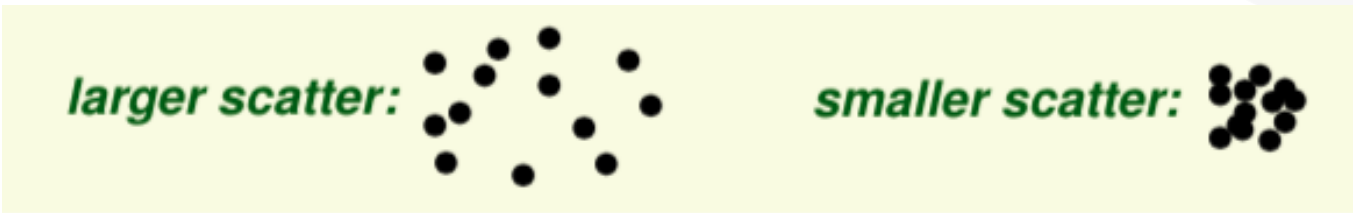
Fisher's Linear Discriminant

Preliminary 2. Introduce scatter

- There are samples $z_1, z_2, ..., z_n$, the sample mean is $\mu_z = \frac{1}{n}\sum_{i=1}^{n} z_i$
- Define samples **scatter** as

$$s = \sum_{i=1}^{n}(z_i - \mu_z)^2$$

- Scatter is just sample variance multiplied by n, it measures the spread of data around the mean.
- Scatter measures the same thing as variance, but on different scale.



Fisher's Linear Discriminant

Let's see how to use mathematical way to present this problem.

- Assume we have a set of D-dimensional samples $X = \{x_1, x_2, ...x_m\}$, N_1 of which belong to class C_1, and N_2 of which belong to class C_2. We also assume the mean vector of two classes in X-space:

$$\mu_k = \frac{1}{N_k} \sum_{i \in C_k} x_i, \quad k = 1, 2.$$

- and in y-space:

$$\hat{\mu}_k = \frac{1}{N_k} \sum_{i \in C_k} \theta^T x_i = \theta^T \mu_k, \quad k = 1, 2$$

Fisher's Linear Discriminant

The **between-class variance** is

- to define a **measure of separation** between two classes is to choose the distance between the projected means

$$\hat{\mu}_2 - \hat{\mu}_1 = \theta^T(\mu_2 - \mu_1)$$

The **within-class variance** for each class C_k is

- use scatter

$$\hat{s}_k^2 = \sum_{i \in C_k} (y_i - \hat{\mu}_k)^2 \quad k = 1, 2$$

Now that we get the between-class variance and within-class variance, we can define our **objective function** $J(\theta)$ as:

$$J(\theta) = \frac{(\hat{\mu}_2 - \hat{\mu}_1)^2}{\hat{s}_1^2 + \hat{s}_2^2}$$

Fisher's Linear Discriminant

Objective function

$$J(\theta) = \frac{(\hat{\mu}_2 - \hat{\mu}_1)^2}{\hat{s}_1^2 + \hat{s}_2^2}$$

If maximizing the objective function $J(\theta)$, we are looking for a projection where

- examples from the class are projected very close to each other (small scatter, which is the denominator)
- the projected means are as farther apart as possible (large difference between two mean value, which is in the numerator)

Fisher's Linear Discriminant

To find the optimum θ^*, we must express $J(\theta)$ as a function of θ.

- 1. The scatter of the projection y can then be expressed as

$$\begin{aligned}\hat{s}_k^2 &= \sum_{i \in C_k} (y_i - \hat{\mu}_k)^2 \\ &= \sum_{i \in C_k} (\theta^T x_i - \theta^T \mu_k)^2 \\ &= \sum_{i \in C_k} \theta^T (x_i - \mu_k)(x_i - \mu_k)^T \theta \\ &= \theta^T S_k \theta\end{aligned}$$

So we can get,

$$\hat{s}_1^2 + \hat{s}_2^2 = \theta^T S_1 \theta + \theta^T S_2 \theta = \theta^T S_W \theta$$

where we denote the class k scatter in feature space-x as:
$S_k = \sum_{i \in C_k} (x_i - \mu_k)(x_i - \mu_k)^T$
within-class scatter matrix: $S_W = S_1 + S_2$

Fisher's Linear Discriminant

- 2. Similarly, the difference between the projected means can be expressed in terms of the means in the original feature space:

$$\begin{aligned}(\hat{\mu}_2 - \hat{\mu}_1)^2 &= (\theta^T \mu_2 - \theta^T \mu_1)^2 \\ &= \theta^T (\mu_2 - \mu_1)(\mu_2 - \mu_1)^T \theta \\ &= \theta^T S_B \theta\end{aligned}$$

where we denote the between-class scatter matrix:
$S_B = (\mu_2 - \mu_1)(\mu_2 - \mu_1)^T$

We can finally express the Fisher criterion in terms of S_W and S_B as:

$$J(\theta) = \frac{\theta^T S_B \theta}{\theta^T S_W \theta}$$

Fisher's Linear Discriminant

Solve the Problem

The easiest way to maximize the object function J is to derive it and set it to zero.
For now, the problem has been solved and we just want to get the direction of the θ, which is the optimum θ^*

$$\theta^* \propto S_W^{-1}(\mu_2 - \mu_1)$$

This is known as Fisher's linear discriminant(1936), although it is not a discriminant but rather a specific choice of direction for the projection of the data down to one dimension, which is $y = \theta^{*T} X$.
[3]

Reference I

[1] Richard O Duda, Peter E Hart, et al. **Pattern Classification**. 2nd ed. Wiley New York, 2000.

[2] Christopher M Bishop. **Pattern Recognition and Machine Learning**. Springer, 2006.

[3] Bingyu Wang Cheng Li. **Fisher Linear Discriminant Analysis**. URL: https://www.ccis.northeastern.edu/home/vip/teach/MLcourse/5_features_dimensions/lecture_notes/LDA/LDA.pdf.

Thank You !

$\mathcal{Q}$ & $\mathcal{A}$