Pattern-Recognition-Review

Mohammad Amin Dadgar

January 2022

Contents

[1] Linear Discriminate Analysis (Fisher's method)

1 Introduction

This paper is the overview of what the Isfahan university students did in their pattern recognition review group.

2 Session 1

In this session we are going to review the first 3 slides of professor Adibi. This session's objectives are:

- What is pattern recognition? challenges in pattern recognition
- How a model works to classify data?
- Preprocess data, training, evaluation, feature extraction, feature selection, Risk of wrong classification. overfitting,
- An overview prior probability, pre- and post probability, bayes theorem, Expected value, variance and covariance
- Gaussian distribution, whitening transformation

2.1 Additional Questions

2.1.1 Question 1

Question:
A classification example was introduced in the course to separate salmon from seabass. In a similar manner, briefly introduce your ideas about the following problems for a three-class pattern classification problem on fruits with banana, orange, and yellow apple as classes: (a) the sensor or sensors to be used, (b) preprocessing problem, (c) segmentation problem, (d) features that can separate these three classes.
Answer:
(a) A set of sensors such as color, shape, fruit skin type can be used.
(b) After reading data from the sensors, they maybe noisy or have missing values or we need to create a new feature combining two features.
(c) having two features shape and color can separate these fruits well.

2.1.2 Question 2

Question:
For a pattern classification problem with two classes, where each class has a Gaussian density function, the obtained decision boundary can sometimes be expressed based on a linear discriminant function. What condition must be satisfied for this case? Briefly explain.
Answer:
See the first and second state for Discriminant Gaussian Function.

first state: having a same variance for all data and uncorrelated features.
second state: Or having the same covariance matrix for all classes.

3 Session 2

In this section we are going to explain the below methods:

- Bayesian Decision rule
- Maximum Likelihood estimation (ML), Maximum a Posterior estimation (MAP)
- Bayesian estimation

3.1 Bayesian Decision rule

To explain the Bayes Decision rule we need to first know what is the loss function and the conditional risk.

3.1.1 Loss function

Loss function can be defined as, The loss that the system need to tolerate when it apply action α_i for class ω_j. We can show the loss function as $\lambda(\alpha_i|\omega_j)$ and for abbreviation form we represent it as λ_{ij}.
Note: We can say that error is a general form of loss function, that the loss function for correct classification (λij for $i = j$) is zero.

3.1.2 Conditional Risk

Conditional risk represent the all loss functions for applying an action for all classes.

$$R(\alpha_i|x) = \sum_{j=1}^{c} \lambda_{ij} p(\omega_j|x) \tag{1}$$

the equation (1), we can say that the risk doing the action α_i, is applying the loss function to each posterior probability.

3.1.3 Bayesian Decision Rule

In Bayes Decision rule we are comparing the conditional risk for every action. We can say that

$$Decide\ \omega_1\ if R(\alpha_1|x) < R(\alpha_2|x)\ otherwise\ \omega_2 \tag{2}$$

3.2 Maximum Likelihood (ML)

Maximum likelihood is a parametric method used to find the best parameter for data distribution. As it is transparent from the name of the method we are going to maximize the likelihood probability of a function. The equation below is to find the maximum likelihood of a probability.

$$p(D|\theta) = argmax \prod_{i=1}^{n} p(x_i|\theta) \tag{3}$$

To maximize the equation above we can find the derivative of $p(D|\theta)$ with respect to θ and make it equal to zero.
Note: We can apply the logarithm function to the equation (3) to make the product as summation.

3.3 Maximum Posterior (MAP)

Maximum posterior is the same as maximum likelihood, but the difference here is we are going to maximize the posterior probability.

$$p(\theta|D) = argmax \prod_{i=1}^{n} p(\theta|x_i) \tag{4}$$

And using the Bayes rule we can expand the equation above as

$$p(\theta|D) = argmax \prod_{i=1}^{n} p(x_i|\theta)p(\theta) \tag{5}$$

Note: Again, we can apply the logarithm function to the equation (2) or (3) to make the product as summation.

3.4 Bayes estimation

In methods such as ML and MAP we were finding a value to maximize a parameter, but in Bayes estimation we are going to find a distribution for the requested parameter.
The steps for Bayes estimation is as below
Step 1: find the posterior $p(\theta|D)$ as

$$p(\theta|D) = \frac{p(D|\theta)p(\theta)}{p(D)} = a \prod_{i=1}^{k} p(x_k|\theta)p(\theta) \tag{6}$$

Step 2: find $p(x|D)$ as

$$p(x|D) = \int p(x|\theta)p(\theta|D)d\theta \tag{7}$$

Step 3: substitute $p(x|D)$ with $p(x|\omega_i, D_i)$ in Bayes rule

$$p(\omega_i|x, D_i) = \frac{p(x|\omega_i, D_i)p(\omega_i)}{\sum_j p(x|\omega_j, D_j)p(\omega_j)} \tag{8}$$

4 Session 3

In this session we are going to explain two dimension reduction methods PCA and LDA.

4.1 PCA

One method to reduce dimensions is Principal Component Analysis or in the abbreviation form PCA. As it is clear from the name of this method we can compute the Principal Components of the data and removing some components will reduce our feature space size. So to have a better view of how this is working we can say that in PCA method we are computing the eigen-vectors of data covariance matrix and removing the eigen-vectors with the lowest eigen-values. The steps below shows how this method works.
step 1: Compute the mean of sample data

$$\bar{x} = \frac{1}{N} \sum_{i=1}^{M} x_i \tag{9}$$

step 2: Normalize the data

$$\phi_i = \frac{x_i - \bar{x}}{\sigma} \tag{10}$$

step 3: Find the covariance matrix using $A = [\phi_1, ..., \phi_M]$

$$C = \frac{1}{M} \sum_{1}^{M} \phi_i \phi_i^t = \frac{1}{M} AA^t \tag{11}$$

step 4: Find eigen-values and eigen-vectors of the covariance matrix
step 5: Sort eigen-vectors with the order of eigen-vectors descending

$$\lambda_1 > \lambda_2 > ... > \lambda_N \tag{12}$$

u are the eigen-vectors corresponding to each eigen-value

$$u_1, u_2, ..., u_n \tag{13}$$

step 6: After finding the eigen-vectors we can represent data using eigen-vectors as the basis set

$$\phi_i = b_1 u_1 + b_2 u_2 + ... + b_N u_N \tag{14}$$

And the coefficients b_i can be calculated as (Note that we also normalize the results)

$$b_i = \frac{u_i^t \phi_i}{u_i^t u_i} \tag{15}$$

step 7 (last step): Dimension reduction, we can reduce dimensions using the eigen-vectors correspond to the biggest eigen-values

$$\hat{\phi} = \sum_{i=1}^{K} b_i u_i \quad K << N \tag{16}$$

So the important thing here is how to choose K. To choose K it's important to keep in mind that the sum of K sorted eigen-values in respect to all need to be more than a threshold (0.9 or 0.95). meaning the equation below

$$\frac{\sum_{i=1}^{K} \lambda_i}{\sum_{i=1}^{N} \lambda_i} > Threshold \tag{17}$$

4.2 LDA [1]

The method PCA was for unsupervised dimension reduction, but to be able to reduce the dimensions with respect to labels (Aka supervised) There is another method named Linear Discriminant Analysis or LDA in abbreviation form. This method tries to reduce the dimension with respect to minimizing the within-class variation and maximizing the between class variation. Some references name this method as fisher's method.

To introduce LDA we need to first get to know what is within-class variation and between-class variation.

4.2.1 Within-Class variation

Within-class variation can be defined as

$$S_w = \sum_{i=1}^{C} \sum_{j=1}^{M_i} (x_j - \mu_i)(x_j - \mu_i)^T \tag{18}$$

4.2.2 Between-Class variation

between-class variation can be defined as

$$S_b = \sum_{i=1}^{C} (\mu_i - \mu)(\mu_i - \mu)^T \tag{19}$$

Note that μ is the mean of entire dataset.

4.2.3 LDA

So now the definition of LDA is clear that the it tries to maximize the between class variation and tries to minimize the within-class variation. Think of the optimal projection is $y = U^T x$, then we can say maximizing the equation below would satisfy the goal

$$J(U) = Tr\{S_w S_b^T\} = Tr\{U^T S_w U)^{-1} (U^T S_b U)\} \tag{20}$$

Note: Because rank of the S_b matrix is $C-1$, the maximum feature space must be $C-1$, and in another way we can say $K \leq C-1$.
In this Session some python code was included.

[1] Linear Discriminate Analysis (Fisher's method)

5 Session 4

In this session we are going to explain Support vector machines (SVM). Support vector machines are used to minimize a function named empirical risk. think of the the discriminant function we had earlier, if we define our data label as z, the equation below is the empirical risk

$$R_{emp}(w, w_0) = \frac{1}{n}\sum_{k=1}^{n}[z_k - g(x_k, w, w_0)]^2 \tag{21}$$

if the empirical risk was zero we had a great classification just on training set, but for test set the accuracy may be low and we can say that the model does not have a good generalization (chance of overfitting).

So we need another method to be the flipped version of the empirical risk. We use VC dimension to achieve our goal. To explain the VC dimension briefly, it can be said that for a count of lines how many points can be always divided. For example think of a linear line, it's clear that the maximum point that can be divided with this line is 3 (Any order of data in 1 dimension), So we can call a one linear line VC dimension as 3.

There are two types of Support Vector Machines, the linearly separable and the one for non-linearly separable data.

5.1 SVM: linearly-separable data

One type of SVM is the SVM that can separate the linearly separable data. the function that shows the separation line is

$$g(x) = w^t x + w_0 \tag{22}$$

Decide ω_1 if $g(x) > 0$ and if $g(x) < 0$ ω_2

To find the answers for separation line margin (ω), we need to calculate these equations

$$\nabla_\lambda(\sum_{i=0}^{n}\lambda_i + \sum_{i,k=0}^{n}\lambda_i\lambda_k z_i z_k x_i^t x_k) = 0 \tag{23}$$

And in equation (23), z is the label for data and lambda are the Lagrangian coefficients. To find w and w_0 in equation (22), we can use these equations below

$$w = \sum_{k=1}^{n}\lambda_k z_k x_k \tag{24}$$

$$w_0 = z_k - w^t x_k \quad (This\, is\, just\, a\, change\, in\, eq.22) \tag{25}$$

$$0 \leq \lambda_i,\ k = 1, 2, ..., n \tag{26}$$

Note: The non-support vectors Lagrangian coefficient is zero.

If we define an error tolerant for some points to be on the wrong side of decision boundary it will be appear in Lagrangian coefficients limitation.

$$0 \leq \lambda_i \leq c,\ k = 1, 2, ..., n \tag{27}$$

5.2 SVM: non-linearly separable data

What if our data was not linearly separable?
To find a solution we need to increase the dimension. The reason behind this is when we increase the dimension the data become more sparse and a linear hyperplain can separate the data.

To increase the data dimension a method is made to make the calculations easier. The method is called Kernel functions. A kernel function can get the inner product of data in much more easier manner. The kernel function must match the mercers condition.

Kernel Function: The kernel function can be any combination of input vectors. A simple type of kernel function can be seen below

$$K(x, y) = (x.y)^d \tag{28}$$

Mercer Condition: Mercers condition nature is an integral but in our work we can say that as below
Step 1: For the data we have find the Gram-matrix (We will call the matrix A)
Step 2: Show that the matrix is positive semi-definite (PSD), by showing the eigen-values are semi-positive (≥ 0), and $AA^T = I$.