Korea Advanced Institute of Science and Technology

School of Electrical Engineering

EE488 Introduction to Machine Learning Spring 2018

Student Name: Dinh Vu

Student ID: 20184187

**Homework 2**

1. **Part 1**
2. **Logistic regression**

Using the regularization penalty where j = {0, 1, 2}, the optimization equation must be solved:

Take derivative with respect to θ then set to 0:

Without regularization penalty, the black decision boundary classifies the training data with no training error as shown in Figure 1.1. This decision boundary cut Ox1 at and cut Ox2 at .

First, let regularize only . Because C is very large, from (1.1) get that then and . The decision boundary will shift up horizontally and become the red line in Figure 1.1. The training error may be increase to 1.

Second, considering only is regularized. Because of the great value of C and (1.1), that mean , the decision boundary will rotate following the opposite direction of the clockwise and nearly become the blue line in Figure 1.1. The training error may be still equal to zero.



**Figure 1.1.** Regularization penalty

Finally, only take the regularization penalty for . Due to the large value of C and equation (1.1), that mean which causes that the decision boundary will rotate following the clockwise and nearly become the green line in Figure 1.1. In this case, the training error increase rapidly.

1. **Fisher’s Linear Discriminative Analysis (FDA)**
2. *Find the maximum of and estimate the projection directions*

It is required to find

From , get

* Mean vector of class:
* Between-class covariance matrix:
* Within-class covariance matrix:

Hence:

* Projection of each data point

**Table 1.1.** Projection direction of each data point

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Data set** | 1, 2 | 2, 3 | 3, 3 | 4, 5 | 5, 5 |
| **Projection** | -0.9862 | -1.0826 | -0.289 | -1.2752 | -0.4817 |

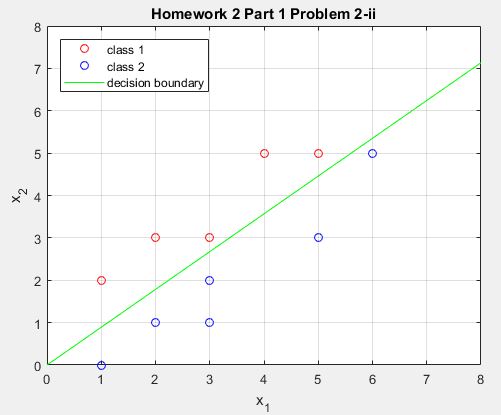
|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Data set** | 1, 0 | 2, 1 | 3, 1 | 3, 2 | 5, 3 | 6, 5 |
| **Projection** | 0.7936 | 0.6972 | 1.4908 | 0.6009 | 1.2982 | 0.3119 |

1. *Plot the data samples*

**Table 1.2.** The output

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Data set** | 1, 2 | 2, 3 | 3, 3 | 4, 5 | 5, 5 |
| **Projection** | -0.9862 | -1.0826 | -0.289 | -1.2752 | -0.4817 |
|  | 1 | 1 | 1 | 1 | 1 |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Data set** | 1, 0 | 2, 1 | 3, 1 | 3, 2 | 5, 3 | 6, 5 |
| **Projection** | 0.7936 | 0.6972 | 1.4908 | 0.6009 | 1.2982 | 0.3119 |
|  | 2 | 2 | 2 | 2 | 2 | 2 |



**Figure 1.2.** The decision line of FDA

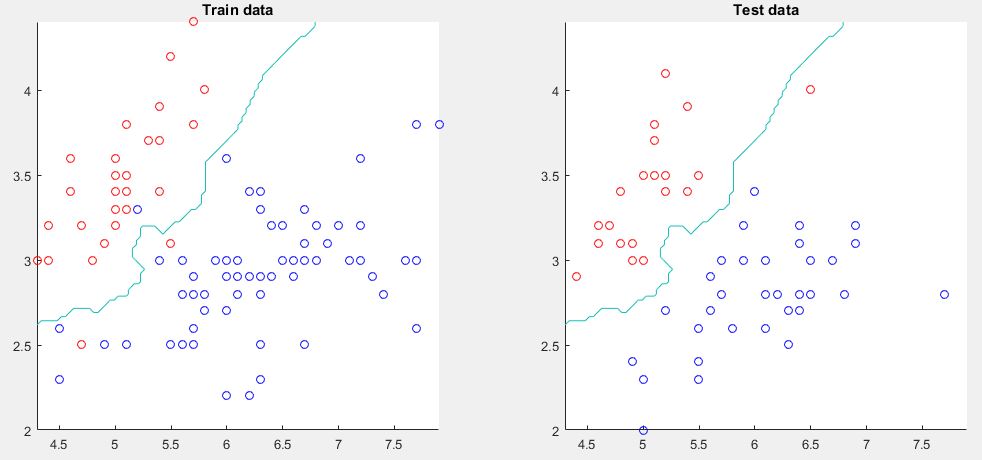
1. **K-fold cross validation**
2. *LOOCV*

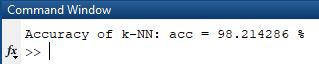
Consider is test data point. Size of the training data set is N – 1. In the training set, the number of label 0 and 1 are and .

According to the given algorithm, the predicted label .

That mean the given algorithm cannot work well in LOOCV model.

1. *K-fold cross validation*
2. **Implementing k-NN algorithm (Matlab Programming)**





**Figure 1.3.** The decision boundary of k-NN algorithm

The result of implementing k-NN algorithm is depicted in the following Figure 1.3. In order to achieve the highest accuracy, the number of the nearest neighbors k should be an odd number and greater than 1 and less than half of the training data size. Therefore, in the line 5 of main\_knn.m file, parameter k is chosen to equal 3 which causes that the accuracy reaches to 98.21 % with the lowest amount of computation.

To programming k-NN algorithm, the distance is calculated by norm 2 and two Matlab functions used: function sort() for searching k nearest training points and function sign() to take major vote.

1. **Part 2**
2. **Linear Discriminative Analysis (LDA)**
3. *Explicit form for f(θ, x)*

* First
* Second
* Third

Because (1) toss a coin with probability of heads to p and (2) the coin toss result in “head” then y = 1 otherwise y = 0, hence:

Replace (2.2) and (2.3) to (2.1), we have:

where:

Finally, is proof

The explicit form for is:

(using ATUB = BTUA)

1. *The difference and similarity between LDA and logistic regression*

* The similarity:
* Both of models are based on probability of sigmoid function
* The model of LDA proofs the assumption of the linear logistic model
* To find hypothesis space, both of them maximize likelihood
* The decision boundary can be any shape and not necessarily linear separator
* In practice, both of methods often give similar results
* The difference:

**Table 2.1.** The difference between LDA and Logistic Regression

|  |  |  |
| --- | --- | --- |
| **Property** | **Linear Discriminative Analysis** | **Logistic Regression** |
| Assumtion about class | No assumption about the probability of class given data p(Ck|x) | Assume that the probability of class given data p(Ck|x) following sigmoid function |
| Distribution of data | Gaussian | Do not know |
| Optimization | Find parameter θ to maximize the joint likelihood of class and data P(x, Ck) | Find parameter θ to maximize the conditional likelihood of class given data p(Ck|x) |
| Noise | Less robust | More robust |
| Training data | Can process the data without class label | Cannot process the data without class label |
| Amount of data | Require less data than Logistic Regression | Need 30% more than LDA to get equally estimation |
| Hypothesis space | It can be any shape | Tend to find a linear separator is it exist |

1. *Decision boundary*
2. μ0 ≠ μ1 and Σ0 = Σ1 = I

The decision boundary is a linear line.

1. μ0 ≠ μ1 and Σ0 =I, Σ1 = 4I

Where:

The decision boundary , where A, B and C are scalar. So it is a circle.

1. μ0 = μ1 = μ and Σ0 =I, Σ1 = 4I

The decision boundary is a circle, similar to the previous b.

The decision boundary is where A is scalar. Therefore, the decision boundary is a hyperbolic.

1. **Kernel SVM**

Assuming that we had , then must be proof.

1. **Kernel SVM**
2. *Soft-margin linear SVM with trade-off factor of C = 0.1*

Corresponding to Figure 4 because the decision boundary is linear and does not separate two classes strictly, which is the case C is small and more errors are allowed.

1. *Soft-margin linear SVM with trade-off factor of C = 10*

This case is for Figure 3 because the decision boundary is linear and separates two class more strictly than in Figure 4, caused by the greater value of C.

1. *Hard-margin kernel machine with*

The decision boundary of quadratic kernel . Because is a second order function of x, the decision boundary can be ellipse or hyperbolic. Hence this kernel is for the hyperbolic curve in Figure 5.

1. *Hard-margin kernel machine with*

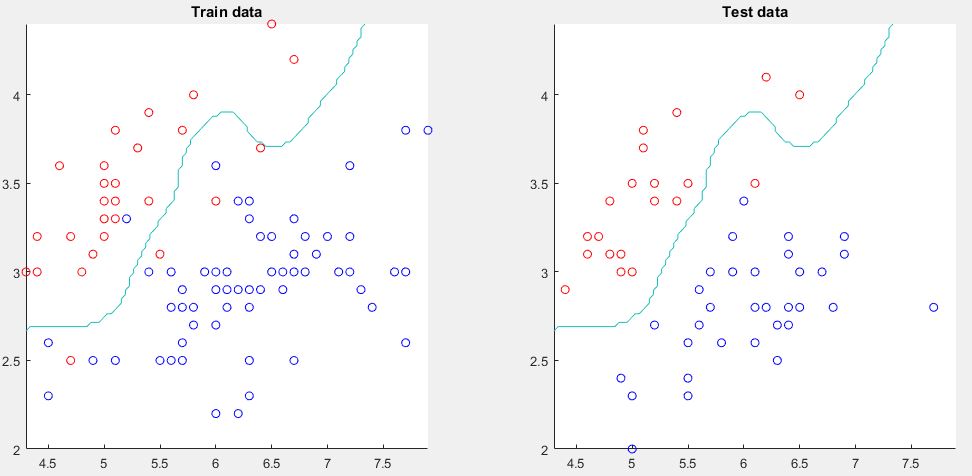
The decision function . If β is very large, then kernel value is small which causes hard classification with the lack of support vectors. Hence it cannot classify many circle point in the middle Figure 1.

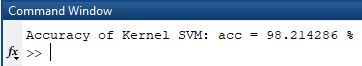
1. *Hard-margin kernel machine with*

Corresponding to Figure 6. Explanation is similar to the previous iv, it can be said that if β is big, there are more support vectors.

1. **Kernel SVM (Matlab Programming)**

The decision boundary, data samples and accuracy is presented in Figure 2.1 below. The source code is in folder kernel\_svm.





**Figure 2.1.** The result of implementing SVM using RBF kernel

To get highest accuracy with lowest execution time, the chosen number of iterations equals to 500, parameter λ = 1 and the covariance of Gaussian distribution σ = 0.2