Assignment 2 – Linear models for classification

Darko Zibar

December 2, 2024

Exercise 1 (40%)
The script `Classification.m` generates three classes, $k = 1, .., 3$. Each class is generated by drawing samples, $\mathbf{x} = [x_1, x_2]$, from a two dimensional Gaussian distribution specified by a mean vector $\boldsymbol{\mu}_k = [\mu_1, \mu_2]$ and a co-variance matrix:

$$\sum\nolimits_k = \begin{bmatrix} \sigma_{11}^2 & \sigma_{12}^2 \\ \sigma_{21}^2 & \sigma_{22}^2 \end{bmatrix} \tag{1}$$

For each of the classed different mean and co-variance matrix is chosen. Each generated sample is then associated with its corresponding class, $\mathcal{C}_k$, where $k = 1, 2$ and 3. For instance if $\mathbf{x}$ is sampled from Gaussian distribution, $\boldsymbol{\mu}_1$ and $\sum_1$, it belongs to class 1, i.e. $\mathcal{C}_1$.

The objective of this exercise is to determine a discriminant function $\mathbf{y}(\mathbf{x}) = \widetilde{\mathbf{W}}^T\widetilde{\mathbf{x}}$ to classify an unknown sample $\mathbf{x}$ to its corresponding class. We start by only considering two classes, i.e. class 1 and 2, i.e. $\mathcal{C}_1$ and $\mathcal{C}_2$.

1. Perform data labelling to generate the data-set. Assign [01] and [10] to Class 1 and 2, respectively. Randomly, shuffle the data.
2. Allocate 80% of the data–set for training and the rest for the testing
3. On the training data–set, employ the least squares method to determine the weight matrix $\widetilde{\mathbf{W}}$
4. Perform classification on the test data–set by computing $\mathbf{y}(\mathbf{x}) = \widetilde{\mathbf{W}}^T\widetilde{\mathbf{x}}$ and assigning the corresponding class
5. Count the number of classification errors

Next, we consider all three classes and the objective is to determine a discriminant function $\mathbf{y}(\mathbf{x}) = \widetilde{\mathbf{W}}^T\widetilde{\mathbf{x}}$ to classify an unknown sample $\mathbf{x}$ to its corresponding class.

1. Modify the data set for two classes to form a data set for three classes. Make sure that you employ the correct class labeling scheme as well as random shuffling of the data within the data set.
2. Allocate 80% of the data–set for training and the rest for the testing
3. On the training data, employ the least squares method to determine the weight matrix $\widetilde{\mathbf{W}}$
4. Perform classification on the test data–set by computing $\mathbf{y}(\mathbf{x}) = \widetilde{\mathbf{W}}^T\widetilde{\mathbf{x}}$ and assigning a class

5. Count the number of classification errors

Exercise 2 (10%)
The objective of this exercise is to demonstrate that the perceptron can be used to learn class labeling and perform accurate classification. It should be kept in mind that the perceptron can only be used for two class problem.

1. Choose the proper class labeling (assignment) and form the data–set. The classes are given by sampling from the two dimensional Gaussian distribution (see Exercise 1)

2. Implement the iterative algorithm for learning the perceptron's weights. The implementation should be performed on the training data–set

3. Evaluate the accuracy of the implementation on the test data–set by plotting the test error

4. Count the number of classification errors

Exercise 3 (20%)
Similar to Exercise 1, consider classification problem involving only two classes. Reuse the code from Exercise 1, to generate two classes and the corresponding data-set for training and test. Implement the logistic regression, (implement the weight update rule), and demonstrate that can it be used to perform accurate classification by computing the classification error. To make the algorithm work vary the step size η and number of iterations. Plot the evolution of weights as a function of iteration number and comment on the results. Vary the variance of the two classes and observe the classification error. Increasing the variance of classes implies more overlap between classes and increased classification error.

Exercise 4 (30%)
In this exercise, we will demonstrate how logistic regression can be used to correctly classify symbols that have been transmitted through a noisy channel.

We assume that transmitter is transmitting a bit sequence of length 10^4, consisting of 1 and 0, i.e. $X_k = \{0, 1\}$ and $k = 1, ..., L$. The transmission channel is noisy. This implies that the channel adds white Gaussian noise to the transmitted bit sequence X_k. The output symbols Y_k are corrupted by noise and thereby diverge from X_k. The noise is modelled as a white Gaussian noise with variance σ_n^2.

1. Plot a histogram of the received signal Y_k when for the noise variance, σ_n^2, of 0.1 and 0.3. What is the impact of increasing noise variance on the received signal?

2. The objective of the receiver is to recover the transmitted bit sequence as accurately as possible from the noisy received sequence Y_k. This implies that we need to learn a decision rule of how to classify Y_k to 0 or 1. Implement a logistic regression based classifier and show that we get zero errors for $\sigma_n^2 = 0$. Plot the evolution of weights as a function of iterations. Allocate 80% of the data for the training.

3. Test how the learned classification model performs on test data. Increase the noise variance, σ_n^2, from 0.1 to 0.4 in steps of 0.01, and compute the miss-classification rate. Plot the results (use log-scale for y-axis). The miss-classification rate is defined as the number or erroneous bits divided by the total number of bits used for testing.