Assignment IV (2015.2 - T01)

Submission Deadline: November, 16th, 2015.

Instructions:

- Submission deadline: November, 16th, 2015.
- This assignment must be delivered as a report (with Introduction, Methodology, Results, Conclusion and References).
- Source codes must be delivered as attachments.
- When answering the questions, books and papers citations are allowed as long as they are listed in "references" section of the report.
- When submitting this assignment, please inform if you are registered or not in SIGAA.
- If you are registered in SIGAA, provide your university ID number when submitting the assignment.

Exercises

Exercise 01

This exercise explores *Least Squares for Classification.* This classifier has a linear model for 2 classes which is given as

$$y(\mathbf{x}) \quad = \quad \mathbf{w}^T\mathbf{x} + w_0. \tag{1}$$

In this exercise, it will be used a data set where the samples are divided in $K = 2$ classes and it is divided in two parts. The first part, called `data-I-samples-learning.txt`, have the samples used for training phase and the second part, called `data-I-labels-learning.txt`, has the labels of each sample. Each row of `data-I-samples-learning.txt` is a sample and each row of `data-I-labels-learning.txt` is this sample's label. For instance, in the 5th row of `data-I-samples-learning.txt` has a sample where

$$\mathbf{x}_5 \quad = \quad \begin{pmatrix} 3.5142 \\ 5.5719 \end{pmatrix}$$

and in the same line of `data-I-labels-learning.txt` (the 5th line) it is seen that

$$t \quad = \quad 1.$$

So the 5th sample has a label $t = 1$ i.e., it belongs to the first class $\mathcal{C}_1$.
Use `data-I-samples-learning.txt` and `data-I-labels-learning.txt` to find the optimum $\mathbf{w}$. Then, test your results using the second part of the dataset (`data-I-samples-testing.txt` and `data-I-labels-testing.txt`. Both follow the same idea of the first part of the data set). For the results:

1.1) Give the sum-of-squares error (E) given as:

$$E \quad = \quad \frac{1}{2}\sum_{n=1}^{N}\left(y_n(\mathbf{x}) - t_n\right)^2,$$

where N is the total number of samples and t_n is the label of the n-th sample;

1.2) Plot the testing data samples along with the decision boundary.

1.3) Give the classification rates (in this case: percentage of correctly classificated samples).

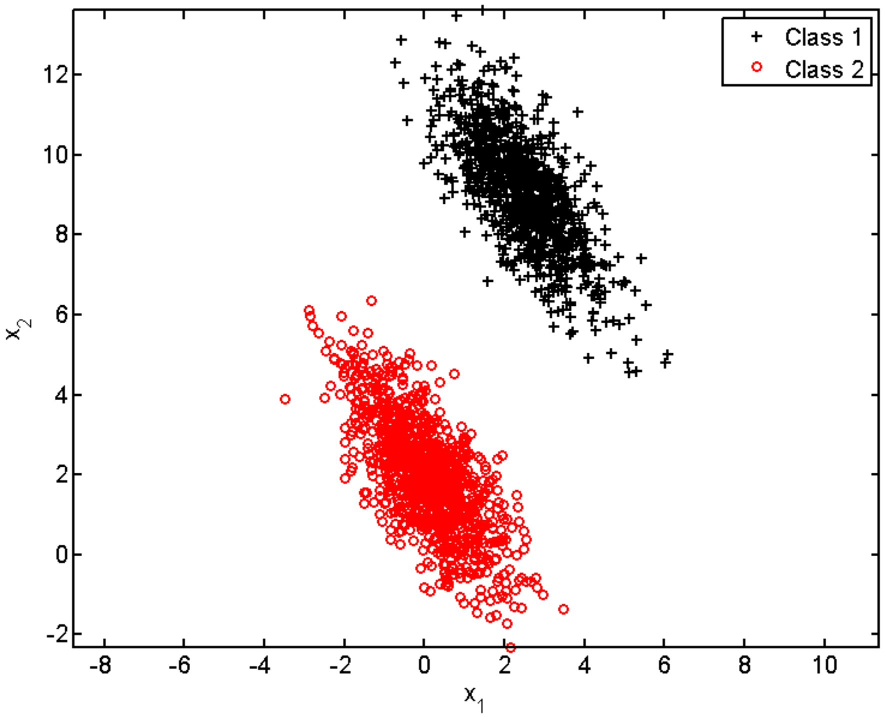

Figure 1: Scatter plot of the data used in **Exercise 02**.

Exercise 02

This exercise explores *Fisher Linear Discriminant.* Generate a data set where it has two classes. The first class, $\mathcal{C}_1$, must have the mean vector

$$\boldsymbol{\mu}_1 = \begin{pmatrix} 2.5 \\ 9 \end{pmatrix}$$

and the second class, $\mathcal{C}_2$, must have the mean vector

$$\boldsymbol{\mu}_2 = \begin{pmatrix} 0 \\ 2 \end{pmatrix}.$$

Both classes must share the same covariance matrix

$$\boldsymbol{\Sigma} = \begin{pmatrix} 1 & -1 \\ -1 & 2 \end{pmatrix}.$$

Each class must have 1000 samples. Figure 1 shows how the scatter plot of the generated data set should look like.

2.1) Use *Fisher's Linear Discriminant*;

2.2) Plot the data *after* the execution of Fisher's Linear Discriminant. Draw the decision boundary.

2.3) Inform the classification accuracy (percentage of correctly classificated samples).

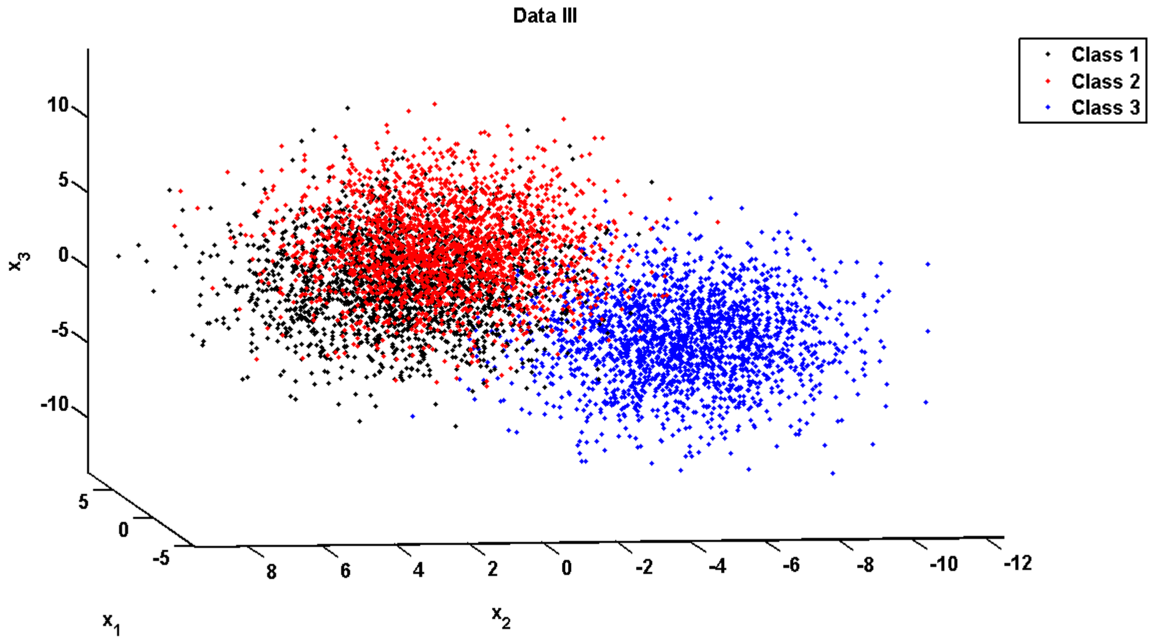

Figure 2: Scatter plot of the data used in **Exercise 04**.

Exercise 03

Using the *Perceptron Algorithm* find the *decision boundary* in the data used in **Exercise 01**. It is possible to find a *decision boundary* with the data available in **Exercise 02**? In both situations, does the *Perceptron Algorithm* give good classification rates (at least 90% of the testing samples classified correctly)? Justify your answer.

Exercise 04

The data set used in this exercise has a *scatter plot* shown in Figure 2. This data set has $K = 3$ classes. Use *Probabilistic Generative Models* in this exercise.

Note that this exercise is using multiclass approach. So consider a generative classification model for K classes defined by prior class probabilities $p(\mathcal{C}_k) = \pi_k$ where

$$\pi_k = \frac{N_k}{N}$$

where N_k is the number of samples assigned to class $\mathcal{C}_k$ and N is the total number of samples. Suppose we are given a training data set $\{\phi_n, \mathbf{t}\}$ where $n = 1, \ldots, N$; $\mathbf{t}_n$ is a binary target vector of length K that that uses 1-of-K coding scheme and $\boldsymbol{\phi}$ is the input feature vector. In this exercise, consider that all classes has the same covariance matrix $\boldsymbol{\Sigma}$ so that

$$p(\boldsymbol{\phi}|\mathcal{C}_k) = \mathcal{N}(\boldsymbol{\phi}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}).$$

The data sets: `data-III-samples-learning.txt` and `data-III-labels-learning.txt` must be used for training phase. In this phase, estimate $\boldsymbol{\mu}_1$ for $\mathcal{C}_1$, $\boldsymbol{\mu}_2$ for $\mathcal{C}_2$ and $\boldsymbol{\mu}_3$ for class $\mathcal{C}_3$. Assume that all classes has the same covariance matrix $\boldsymbol{\Sigma}$.
Use maximum likelihood solution for the mean

$$\boldsymbol{\mu}_k = \frac{1}{N_k}\sum_{n=1}^{N} t_{nk}\boldsymbol{\phi}_n$$

and for the covariance matrix

$$\begin{aligned}\mathbf{\Sigma} &= \sum_{k=1}^{K}\frac{N_k}{N}\mathbf{S}_k,\\ \mathbf{S}_k &= \frac{1}{N_k}\sum_{n=1}^{N} t_{nk}(\boldsymbol{\phi}_n-\boldsymbol{\mu}_k)(\boldsymbol{\phi}_n-\boldsymbol{\mu}_k)^T\end{aligned}$$

where t_{nk} is the element of the identity matrix where is in n-th row and k-th column. After building the whole model, use `data-III-samples-testing.txt` and `data-III-labels-testing.txt` to calculate $p(\mathcal{C}_1|\mathbf{x})$ and $p(\mathcal{C}_2|\mathbf{x})$. Plot the *decision boundaries* (hyperplanes) along with the scatter-plot of the data. Inform the classification rates.

Exercise 05

With the same data used in **Exercise 04**, use multiclass logistic regression and compare the results with the algorithm used in the previous exercise. Recall that in this problem, $\mathbf{t}_n$ uses the 1-of-K coding scheme is used in the target vector $\mathbf{t}_n$. The likelihood function is given by

$$\begin{aligned}p\left(\mathbf{T}|\mathbf{w}_1,\ldots,\mathbf{w}_K\right) &= \prod_{n=1}^{N}\prod_{k=1}^{K} p(\mathcal{C}_k|\boldsymbol{\phi}_n)^{t_{nk}}\\ &= \prod_{n=1}^{N}\prod_{k=1}^{K} y_{nk}^{t_{nk}}\end{aligned}$$

where $y_{nk} = y_k(\boldsymbol{\phi}_n)$ and $\mathbf{T}\in\mathbb{R}^{N\times K}$ is a matrix of target variables with elements t_{nk}. Taking the negative logarithm then gives

$$\begin{aligned}E(\mathbf{w}_1,\ldots,\mathbf{w}_K) &= -\ln\left(p(\mathbf{T}|\mathbf{w}_1,\ldots,\mathbf{w}_K)\right)\\ &= -\sum_{n=1}^{N}\sum_{k=1}^{K} t_{nk}\ln(y_n k)\end{aligned}$$

which is knows as the *cross-entropy* error function for the multiclass classification problem.

$$\nabla_{\mathbf{w}_k}\nabla_{\mathbf{w}_j}E(\mathbf{w}_1,\ldots,\mathbf{w}_K) = -\sum_{n=1}^{N} y_{nk}\left(I_{kj}-y_{nj}\right)\boldsymbol{\phi}_n\boldsymbol{\phi}_n^T.$$

As with the two-class problem, the Hessian matrix for the multiclass logistic regression model is positive definite and so the error function has a unique minimum.
Multi class IRLS algorithm is given by

$$\mathbf{w}_k^{\text{new}} = \left(\mathbf{\Phi}^T\mathbf{R}_k\mathbf{\Phi}+\lambda\mathbf{I}\right)^{-1}\mathbf{\Phi}^T\mathbf{R}_k\mathbf{z}_k,$$

where $\mathbf{R}_k$ is a diagonal matrix with elements in the diagonal are $y_{nk}(1-y_{nk})$, λ is a small number between 0 and 1, $\mathbf{I}$ is the identity matrix and

$$\mathbf{z}_k = \mathbf{\Phi}\mathbf{w}_k^{\text{old}} - \left(\mathbf{R}_k\right)^{-1}\left(\mathbf{y}_k-\mathbf{t}_k\right)$$

where $\mathbf{y}_k$ and $\mathbf{t}_k$ are respectively the output and target vector for the k-th class.

Exercise 06

In this course's website there is a data set called `iris.dat` taken from UCI Machine Learning Repository [1]. The data set is has 3 classes where each of these classes have 50 samples. Each sample has 4 attributes. The last column of the data set are the labels of each sample. More information about `iris.dat` is available at
`https://archive.ics.uci.edu/ml/datasets/Iris`.

6.1) Inform range, variance and standard deviaton of each attribute.

6.2) Use Principal Component Analysis (PCA) to reduce the dimensionality of the data to 2-D. Use the following PCA algorithm:

I) Remove the mean from the data.

II) Divide the data by its standard deviation.

III) Put all the data in a matrix $\mathbf{X} \in \mathbb{R}^{N \times M}$.

IV) Compute $\mathbf{S} = \frac{1}{N-1}\mathbf{X}^T\mathbf{X}$ where N is the total number of samples (in this case: $N = 150$).

V) Compute the eigen-value decomposition $\mathbf{S} = \mathbf{P}\mathbf{\Lambda}\mathbf{P}^T$, such that

$$\mathbf{\Lambda} = \begin{pmatrix} \lambda_1 & 0 & \dots & 0 \\ 0 & \lambda_2 & \dots & 0 \\ 0 & 0 & \ddots & 0 \\ 0 & 0 & \dots & \lambda_M \end{pmatrix},$$

where $\lambda_1 > \lambda_2 > \dots > \lambda_M$ are the eigenvalues of $\mathbf{S}$. The matrix $\mathbf{P} \in \mathbb{R}^{M \times N}$ is organized as

$$\mathbf{P} = (\mathbf{p}_1|\mathbf{p}_2|\dots|\mathbf{p}_M),$$

where $\mathbf{p}_i$ is the i-th eigenvector associated with the i-th eigenvalue.

VI) Construct a matrix $\tilde{\mathbf{P}}$ where is made of the two first columns of $\mathbf{P}$

$$\tilde{\mathbf{P}} = (\mathbf{p}_1|\mathbf{p}_2)$$

VII) Compute the score matrix

$$\mathbf{T} = \mathbf{X}\tilde{\mathbf{P}}$$

and use this new matrix $\mathbf{T}$ as 2-D data.

6.3) Find the decision boundary (in the new data) using Least Squares for Classification.

-

Good luck!

References

[1] M. Lichman. UCI machine learning repository, 2013.