Korea Advanced Institute of Science and Technology
School of Electrical Engineering
**EE488 Intro. to Machine Learning**, Spring 2018

Issued: Apr. 6, 2018
Due: Apr. 13, 2018

# Assignment II-part1

---

## Policy

Group study is encouraged; however, assignment that you hand-in must be of your own work. Anyone suspected of copying others will be penalized. The homework will take considerable amount of time so start early.

1. **Logistic regression**: We consider here a discriminative approach for solving the classification problem illustrated in Figure 1. We attempt to solve the binary classification task depicted in

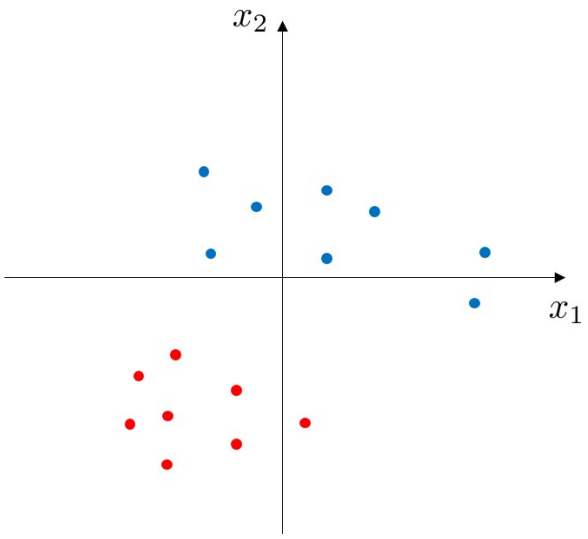



Figure 1: The 2-dimensional labeled training set, where blue dots corresponds to class $y = 1$ and red dots corresponds to class $y = 0$.

Figure 1 with the simple linear logistic regression model

$$p(y = 1|\boldsymbol{x}, \boldsymbol{\theta}) = g(\theta_0 + \theta_1 x_1 + \theta_2 x_2) = \frac{1}{1 + exp(-\theta_0 - \theta_1 x_1 - \theta_2 x_2)}$$

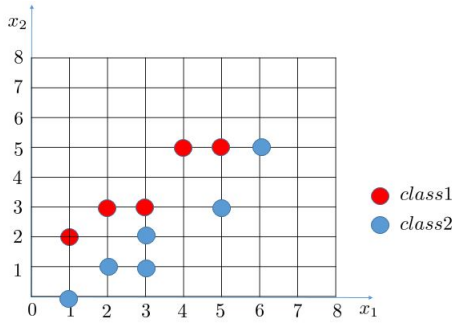
Notice that the training data can be separated with zero training error with a linear separator. Consider training regularized linear logistic regression models where we try to maximize

$$\sum_{i=1}^{n} \log(P(y_i|x_i, \theta_0, \theta_1, \theta_2)) - C\theta_j^2$$

for very large $C$. The regularization penalty used in penalized conditional log-likelihood estimation is $-C\theta_j^2$ where $j = \{0, 1, 2\}$. In other words, only one of the parameters is regularized

in each case. Given the training data in Figure 1, Draw the decision line with regularization of each parameter $\theta_j$. State whether the training error increases or stays the same (zero) for each $\theta_j$ for very large $C$. Provide a brief justification for each of your answers.

2. **Fisher's Linear Discriminative Analysis (FDA)**:
Given 11 data sets {([1, 2], 1), ([2, 3], 1), ([3, 3], 1), ([4, 5], 1), ([5, 5], 1), ([1, 0], 2), ([2, 1], 2), ([3, 1], 2), ([3, 2], 2), ([5, 3], 2), ([6, 5], 2)} where each data set represents a feature $\boldsymbol{x}_n \in \mathbb{R}^2$ and a label $y_n \in \{1, 2\}$ respectively as shown in Figure. Fisher's linear discriminant is to project on line in the direction $\boldsymbol{\theta}$ which maximizes $J(\boldsymbol{\theta}) = \frac{(m_2 - m_1)^2}{s_1^2 + s_2^2} = \frac{\boldsymbol{\theta}^T \mathbf{S}_B \boldsymbol{\theta}}{\boldsymbol{\theta} \boldsymbol{S}_\theta \boldsymbol{\theta}}$ where $m_i, s_i^2$ are the projected mean and the variance of $i$th class respectively, between-class covariance matrix $\boldsymbol{S}_B = (\boldsymbol{m}_2 - \boldsymbol{m}_1)(\boldsymbol{m}_2 - \boldsymbol{m}_1)^T$, within-class covariance matrix $\boldsymbol{S}_\theta = \sum_{n \in C_1} (\boldsymbol{x}_n - \boldsymbol{m}_1)(\boldsymbol{x}_n - \boldsymbol{m}_1)^T + \sum_{n \in C_2} (\boldsymbol{x}_n - \boldsymbol{m}_1)(\boldsymbol{x}_n - \boldsymbol{m}_2)^T$




   (i) Set $\frac{\partial J(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}} = 0$ to find the maximum of $J(\boldsymbol{\theta})$ and estimate the projection directions $\boldsymbol{\theta}$.

   (ii) Scatters plots the data samples and computes output $y_n = \boldsymbol{\theta}^T \boldsymbol{x}_n$ and represents class 1 samples in red while the class 2 samples in blue.

3. $K$**-fold cross validation**: To evaluate a prediction model we use $K$-fold cross validation method. First we split dataset into $K$ roughly equal sized sets and we fit the model for $K - 1$ sets of data and test it for $k^{th}$ set. We do this for $k = 1, 2, ..., K$ and take the average. When $K = N$($N$ is number of data), it is known as leave-one-out cross validation(LOOCV).

   (i) Consider a case in that the label is chosen at random according to $\mathbb{P}[y = 1] = \mathbb{P}[y = 0] = 1/2$. Consider a learning algorithm that outputs the constant predictor $h(\mathbf{x}) = 1$ if the parity of the labels on the training set is 1 and otherwise the algorithm output the constant predictor $h(\mathbf{x}) = 0$. This case is known as failure case of LOOCV, explain why.

   (ii) Assume above case, this time use $K$-fold cross validation($K \neq N$) instead of LOOCV. Does it fails again, explain you answer.

4. **Implementing $k$-NN algorithm(Matlab Programming Assignment)** In this problem, we will implement a $k$-NN algorithm. The following equation is the mathematical representation for $k$-NN algorithm where given data is noted as $(\mathbf{x_i}, y_i)$, and the Algorithm 1 explains step by step procedure for $k$-NN algorithm.

$$\hat{f}(\mathbf{x}) = \text{argmax}_t \Big( \sum_{i \in N_K(\mathbf{x})} \mathbb{1}(t = y_i) | N_K(\mathbf{x}) : \text{A set of } k \text{ nearest data to } \mathbf{x} \Big)$$

**Algorithm 1:** $k$-NN algorithm

**Goal** : Predict class $y$
**Parameter** : $k$
**for** $t = 1, ..., T$ **do**
  Calculate distance($d_t$) between input $\mathbf{x}$ and data $\mathbf{x}_t$
**end**
Choose $k$ number of samples that are nearest from the input $\mathbf{x}$
Predict class into the majority class
**Output** : Majority voted class $y$

For implementation, a skeleton code is provided in the "knn" folder. The main function `main_knn.m` calls the following 4 functions:(1) `[x_train, y_train, x_test, y_test] = createDataset ('train.csv', 'test.csv')`, (2) `[class] = classify_knn(x_test,x_train,y_train,k)`,(3) `[acc] = test_knn(class,y_test)`, (4) `plot_knn(x_train,y_train,x_test,y_test,k)`.

(i) `[x_train,y_train,x_test,y_test] = createDataset('train.csv','test.csv')` loads both the training and test dataset contained respectively in `train.csv` and `test.csv`.

(ii) `[class] = classify_knn(x_test,x_train,y_train,k)` takes both the training set {`x_train,y_train`} where `x_train` is the data array while `y_train` is the corresponding label array of training dataset and test set `x_test` as inputs to the $k$-NN and outputs the predicts class of test data `class`.

(iii) `[acc]=test_knn(class,y_test)` takes test label `y_test`, predicted class `class` and estimates the classification accuracy in `acc`.

(iv) `plot_knn(x_train,y_train,x_test,y_test,k)` takes the training set {`x_train,y_train`} to scatter plots the loaded data and draw a decision boundary for training set. The positive samples should be represented as red dots while the negative samples should be in blue dots.

# Submit Instructions for Programming Assignment

- Please submit in .zip file to KLMS named **ee488_assignment2_student#.zip**, for example, "ee488_assignment2_20181234.zip".

- This file should contain three folders - **knn** and each folder contains document file for the result with analysis.

- In matlab code, the comment explaining your code **must be** included, or you will not get a full grade even if your code works fine. Please also include all the files that are required to run the code in the zip file. Do not change the name of the folder and comments should be written in English. Additionally submitting unexecutable code will receive no points.