CSE474/574 Introduction to Machine Learning
Programming Assignment 3
Classification and Regression

Due Date: May 1st 2015
Maximum Score: 100

Note

A zipped file containing skeleton Python script files and data is provided. Note that for each problem, you need to write code in the specified function withing the Python script file. **For logistic regression, do not use any Python libraries/toolboxes, built-in functions, or external tools/libraries that directly perform the learning or prediction.**. Using any external code will result in 0 points for that problem.

Evaluation

We will evaluate your code by executing `script.py` file, which will internally call the problem specific functions. Also submit an assignment report (pdf file) summarizing your findings. In the problem statements below, the portions under REPORT heading need to be discussed in the assignment report.

1 Introduction

In this assignment, we will extend the first programming assignment in solving the problem of handwritten digit classification. In particular, your task is to implement Logistic Regression and use the Support Vector Machine tool in `sklearn.svm.SVM` to classify hand-written digit images and compare the performance of these methods.
To get started with the exercise, you will need to download the supporting files and unzip its contents to the directory you want to complete this assignment.

1.1 Datasets

In this assignment, we still use the same data set of first programming assignment - MNIST. In the script file provided to you, we have implemented a function *preprocess()* with preprocessing steps (apply feature selection and feature normalization) and divide data set into 3 parts: training set, validation set and testing set.

2 Your tasks

- Implement **Logistic Regression** and give the prediction results.

- Use **Support Vector Machine (SVM)** toolbox to perform classification.

- Write a report to explain the experimental results with these 2 methods.

2.1 Logistic Regression

2.1.1 Formula derivation

Suppose that a given vector $\mathbf{x} \in \mathbb{R}^D$ is any input vector and we want to classify x into correct class C_1 or C_2. In Logistic Regression, the posterior probability of class C_1 can be written as follow:

$$y = P(C_1|\mathbf{x}) = \sigma(\mathbf{w}^T\mathbf{x} + w_0)$$

where $\mathbf{w} \in \mathbb{R}^D$ is the weight vector.

For simplicity, we will denote $\mathbf{x} = [1, x_1, x_2, \cdots, x_D]$ and $\mathbf{w} = [w_0, w_1, w_2, \cdots, w_D]$. With this new notation, the posterior probability of class C_1 can be rewritten as follow:

$$P(C_1|\mathbf{x}) = \sigma(\mathbf{w}^T\mathbf{x}) \tag{1}$$

And posterior probability of class C_2 is:

$$P(C_2|\mathbf{x}) = 1 - P(C_1|\mathbf{x})$$

We now consider the data set $\{\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_N\}$ and corresponding label $\{t_1, t_2, \cdots, t_N\}$ where

$$t_i = \begin{cases} 1 & \text{if } \mathbf{x}_i \in C_1 \\ 0 & \text{if } \mathbf{x}_i \in C_2 \end{cases}$$

for $i = 1, 2, \cdots, N$.

With this data set, the likelihood function can be written as follow:

$$p(\mathbf{t}|\mathbf{w}) = \prod_{n=1}^{N} y_n^{t_n}(1 - y_n)^{1-t_n}$$

where $y_n = \sigma(\mathbf{w}^T\mathbf{x}_n)$ for $n = 1, 2, \cdots, N$.

We also define the error function by taking the negative logarithm of the log likelihood, we gives the cross-entropy error function of the form:

$$E(\mathbf{w}) = -\ln p(\mathbf{t}|\mathbf{w}) = -\sum_{n=1}^{N}\{t_n \ln y_n + (1 - t_n)\ln(1 - y_n)\} \tag{2}$$

The gradient of error function with respect to w can be obtained as follow:

$$\nabla E(\mathbf{w}) = \sum_{n=1}^{N}(y_n - t_n)\mathbf{x}_n \tag{3}$$

Up to this point, we can use again - gradient descent - to find the optimal weight $\hat{\mathbf{w}}$ to minimize the error function with the formula:

$$\mathbf{w}^{new} = \mathbf{w}^{old} - \gamma\nabla E(\mathbf{w}^{old}) \tag{4}$$

2.1.2 Implementation

You are asked to implement Logistic Regression to classify hand-written digit images into correct corresponding labels. In particular, you have to build 10 binary-classifiers (one for each class) to classify that class from all other classes. In order to implement Logistic Regression, you have to complete function *blrObjFunction()* provided in the base code (*script.py*). The input of *blrObjFunction.m* includes 3 parameters:

- **X** is a data matrix where each row contains a feature vector in original coordinate (not including the bias 1 at the beginning of vector). In other words, $\mathbf{X} \in \mathbb{R}^{N \times D}$. So you have to add the bias into each feature vector inside this function. In order to guarantee the consistency in the code and utilize automatic grading, please add the bias at the beginning of feature vector instead of the end.

- **w** is a column vector representing the parameters of Logistic Regression. Size of **w** is $(D+1) \times 1$.

- **t** is a column vector representing the labels of corresponding feature vectors in data matrix **X**. Each entry in this vector is either 1 or 0 to represent whether the feature vector belongs to a class C_k or not ($k = 1, 2, \cdots, 10$). Size of **t** is $N \times 1$ where N is the number of rows of **X**.

Function *blrObjFunction()* has 2 outputs:

- *error* is a scalar value which is the result of computing equation (2)

- **error_grad** is a column vector of size $(D+1) \times 1$ which represents the gradient of error function obtained by using equation (3).

For prediction using Logistic Regression, given 10 weights vector of 10 classes, we need to classify a feature vector into a certain class. In order to do so, given a feature vector **x**, we need to compute the posterior probability $P(C_k|\mathbf{x})$ and the decision rule is to assign **x** to class C_k that maximizes $P(C_k|\mathbf{x})$. In particular, you have to compltete the function *blrPredict()* which returns the predicted label for each feature vector. Concretely, the input of *blrPredict()* includes 2 parameters:

- Similar to function *blrObjFunction()*, **X** is also a data matrix where each row contains a feature vector in original coordinate (not including the bias 1 at the beginning of vector). In other words, **X** has size $N \times D$. In order to guarantee the consistency in the code and utilize automatic grading, please add the bias at the beginning of feature vector instead of the end.

- **W** is a matrix where each column is a weight vector of classifier k. Concretely, **W** has size $(D+1) \times K$ where $K = 10$ is the number of classifiers.

The output of function *blrPredict()* is a column vector **label** which has size $N \times 1$, in which a feature vector is classify to digit k will have label $k + 1$.

2.2 Support Vector Machines

In this part of assignment, you are asked to use the Support Vector Machine tool in `sklearn.svm.SVM` to perform classification on our data set. The details about the tool are provided here: `http://scikit-learn.org/stable/modules/generated/sklearn.svm.SVC.html`.

Your task is to fill the code in Support Vector Machine section of *script.py* to learn the SVM model and compute accuracy of prediction with respect to training data, validation data and testing using the following parameters:

- Using linear kernel (all other parameters are kept default).

- Using radial basis function with value of gamma setting to 1 (all other parameters are kept default).

- Using radial basis function with value of gamma setting to default (all other parameters are kept default).

- Using radial basis function with value of gamma setting to default and varying value of C $(1, 10, 20, 30, \cdots, 100)$ and plot the graph of accuracy with respect to values of C in the report.

3 Submission

You are required to submit a single file called *proj3.zip* using UBLearns.
File *proj3.zip* must contain 3 files: *report*, *params.pickle* and *script.py*. The *params.pickle* file should contain the weight matrix,**W**, learnt for the logistic regression

- Submit your report in a pdf format. Please indicate the team members, group number, and your course number on the top of the report.

- The code file should contain all implemented functions. Please do not change the name of the file.

Project report: The hard-copy of report will be collected in class at due date. Your report should include the experimental results you have performed using Logistic Regression and Support Vector Machine.

4   Grading scheme

- Implementation:

    - ✓ *blrObjFunction()*: 20 points
    - ✓ *blrPredict()*: 20 points
    - ✓ *script.py*: 20 points (your code in SVM section)

- Project report: 30 points

- Accuracy of classification methods: 10 points