Stats 141C	**High Performance Statistical Computing**	**Spring 2018**

Homework 1

Lecturer: Cho-Jui Hsieh   Date Due: May 1, 10:20pm (in class), 2018

Keywords: *Classification, Regression.*

For this homework, we will try classification and regression using datasets listed in `https://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/`. The format of data is called "SVMLight" format:

```
<label> <index>:<value1> <index>:<value2> . . .
```

Each line contains an instance. For classification, <label> is an integer indicating the class label. For regression, <label> is the target value which can be any real number. Indices are in ascending order. This is another way to present a sparse data matrix; the missing indices are zeros. Consider using "sklearn.datasets.load_svmlight_file" function to load data or you can also write the data reading program using standard Python I/O text file reading.

Problem 1. Regression [60 pt]

1. Download the "cpusmall" data from `https://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/regression.html#cpusmall`. Solve the ridge regression problem:

$$\boldsymbol{w}^* = \arg\min_{\boldsymbol{w}} \{ \frac{1}{n} \sum_{i=1}^{n} (\boldsymbol{x}_i^T \boldsymbol{w} - y_i)^2 + \frac{\lambda}{2} \|\boldsymbol{w}\|^2 \} := f(\boldsymbol{w}), \tag{1}$$

   where $\boldsymbol{x}_i \in \mathbb{R}^d$ is the i-th training sample, and $y_i \in \mathbb{R}$ is the i-th target value.

   Write the gradient descent algorithm with fixed step size for solving (1). The gradient descent algorithm is in Algorithm 1. Set $\lambda = 1, \epsilon = 0.001$ and test the algorithm for different step sizes: $\eta = 10^{-7}, 10^{-6}, 10^{-5}, 10^{-4}, 10^{-3}, 10^{-2}$. Report your findings.

2. Set $\lambda = 1$ and do 5-fold cross validation—randomly split data into 5 subsets; each time using 4 of them as training and 1 of them as testing to get the Mean Square Error (MSE), and then report the average MSE. The MSE is defined by

$$\frac{1}{n_{test}} \sum_{i=1}^{n_{test}} (\boldsymbol{x}_i^T \boldsymbol{w} - y_i)^2,$$

   where n_{test} is the number of testing instances.

3. Run gradient descent (with fixed step size) on "E2006-tfidf" data (see `https://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/regression.html#E2006-tfidf`). The training data can be downloaded from `https://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/regression/E2006.train.bz2`. The testing data can be downloaded from `https://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/regression/E2006.test.bz2`. Download both training and testing data from the website. Run your gradient descent implementation. Set $\epsilon = 0.001$ and $\lambda = 1$, try to select a good step size, and report your step size and the test MSE you get.

Algorithm 1 Gradient Descent with Fixed Step Size

- Input: η: step size, ϵ: Stopping condition, $\boldsymbol{w}_0$: initial solution
- $\boldsymbol{w} \leftarrow \boldsymbol{w}_0$
- $r_0 \leftarrow \|\nabla f(\boldsymbol{w}_0)\|$
- For iter $= 1, 2, \ldots, 200$ (Maximum 200 iterations)
  - $\boldsymbol{g} = \nabla f(\boldsymbol{w})$
  - If $\|\boldsymbol{g}\| \leq \epsilon r_0$: Break (End program)
  - $\boldsymbol{w} \leftarrow \boldsymbol{w} - \eta \boldsymbol{g}$

Problem 2. Classification (Logistic Regression) [40pt]

In this problem, you will write your own code for logistic regression. Given training data $\{\boldsymbol{x}_i, y_i\}$ for $i = 1, 2, \cdots, n$. Each $\boldsymbol{x}_i$ is a feature vector and each y_i is the +1/-1 label. Logistic regression model can be learned by solving

$$\boldsymbol{w}^* = \arg\min_{\boldsymbol{w}} \{ \frac{1}{n} \sum_{i=1}^{n} \log(1 + e^{-y_i \boldsymbol{w}^T \boldsymbol{x}_i}) + \frac{\lambda}{2} \|\boldsymbol{w}\|^2 \} := f(\boldsymbol{w}). \quad (2)$$

The model $\boldsymbol{w}^*$ can then be used for prediction.

1. Derive the gradient of (2).
2. Implement gradient descent with fixed step size to solve (2). Test it on news20 binary classification dataset (`https://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/binary/news20.binary.bz2`). Note that there will be only one file (news20.binary). Split it into 80% training and 20% testing. Solve the logistic regression problem using $\lambda = 1$ on the training set, and report the prediction accuracy on test set.