Boston University
Department of Electrical and Computer Engineering

ENG EC 503 (Ishwar) Learning from Data

Assignment 5

Issued: Fri 10 Mar 2017 **Due:** 5pm Wed 22 Mar 2017 in box outside PHO440 + Blackboard
Required reading: Your notes from lectures and additional notes on website on logistic regression.

- **Advise**: This homework assignment requires a **large** amount of time and effort. We urge you to start right away. This assignment cannot be finished by staying up all night just before the deadline.
- This homework assignment requires programming in MATLAB. If you are new to MATLAB programming, please refer to the following link for a primer:
`http://www.math.ucsd.edu/~bdriver/21d-s99/matlab-primer.html`
- You will be making two submissions: (1) A paper submission in the box outside PHO440. (2) An electronic submission of all your matlab code to blackboard (in a single zipped file appropriately named as described below).
- **Paper submission:** This must include all plots, figures, tables, numerical values, derivations, explanations (analysis of results and comments), and also printouts of all the matlab .m files that you either created anew or modified. Submit color printouts of figures and plots whenever appropriate. Color printers are available in PHO305 and PHO307. Be sure to annotate figures, plots, and tables appropriately: give them suitable *titles* to describe the content, label the *axes*, indicate *units* for each axis, and use a *legend* to indicate multiple curves in the plots. Please also explain each figure properly in your solution.
- **Blackboard submission:** All the matlab .m files (and only .m files) that you either create anew or modify must be appropriately named and placed into a **single** directory which should be zipped and uploaded into the course website. Your directory must be named as follows:
`<yourBUemailID>_assignment5`. For example, if your BU email address is `mary567@bu.edu` you would submit a single directory named: `mary567_assignment5.zip` which contains all the matlab code (and only the code).
- **File naming convention:** Instructions for file names to use are provided for each problem. As a general rule, each file name must begin with your BU email ID, e.g., `mary567_<filename>.m`. The file name will typically contain the problem number and subpart, e.g., for problem 5.1b, the file name would be `mary567_assignment5_1b.m`. Note that the dot . in 5.1 is replaced with an underscore (this is important).

Problem 5.1 San Francisco Crime Prediction: (∼1–2hrs run time)
In this problem we will work with a recent real-world dataset that can be downloaded from the following website: `https://www.kaggle.com/c/sf-crime`. This dataset is hosted by Kaggle as part of a competition for the machine learning community to use for fun and practice. The competition started at 8:51 pm on Tuesday 2 June 2015 UTC and ended at 11:59 pm on Monday 6 June 2016 UTC (370 total days). The dataset contains information about various incidents ranging from January 2003 to March 2015 derived

from the San Francisco Police Department Crime Incident Reporting system. The raw data has been pre-processed into two MATLAB .mat files: `data_SFcrime_train.mat` and `data_SFcrime_test.mat`. Each incident contains the following attributes:

- **Dates**: The times at which the crimes occurred. We will only extract the "Hour" (in 24-hour format).
- **Category**: The type of crime. This is the class label and is only available for the training set.
- **DayOfWeek**: Weekday of crime: Sunday, Monday, Tuesday, etc.
- **PdDistrict**: Police department district.
- **Address**: Street address of crime. We are not going to use this attribute in this MATLAB exercise.
- **X** and **Y**: GPS coordinates (**X** = Longitude, **Y** = Latitude) of the crime location. We are not going to use this attribute in this MATLAB exercise. You can, however, use them to visualize the incidents on a map of San Francisco.

(a) In this exercise we are going to make use of Hour (extracted from Date), DayOfWeek, and PdDistrict to build a classifier. We will treat all of them as *categorical* variables even though some of them like Hour and DayOfWeek have some quantitative and ordinal aspects to them. We will convert these three categorical variables into real-valued vectors as follows. The variable DayOfWeek will be represented as a 7-dimensional vector with "Sunday"= $[1, 0, 0, 0, 0, 0, 0]$, "Monday"= $[0, 1, 0, 0, 0, 0, 0]$, and so on. Similarly, the Hour variable will be represented as a 24-dimensional vector of zeros with a single one in the slot corresponding to the hour. The PdDistrict variable will likewise be a 10-dimensional vector of zeros with a single one in the slot corresponding to the police district (there are 10 police districts in the dataset). We will then concatenate all three of these vectors (Hour, DayOfWeek, PdDistrict) into one long binary feature vector of length $24 + 7 + 10 = 41$. Perform these pre-processing steps for both the training and the test dataset.

(i) **Submit** your script for processing the categorical features with name `<yourBUemailID>_assignment5_1a.m`.

(ii) **Plot** three histograms (one for Hour, one for DayOfWeek, and one for PdDistrict) for all the incidents contained in the file `data_SFcrime_train.mat`.

(iii) Based on the incidents in `data_SFcrime_train.mat`, **find and report** the most likely hour of occurrence of each type of crime.

(iv) Based on the incidents in `data_SFcrime_train.mat`, **find and report** the most likely type of crime within each PdDistrict.

Visualization aid: For the purpose of visualization, we have provided you with a map visualization tool: `plot_google_map.m`. If you discover any interesting patterns or create any informative figures for visualization, please include them in your report.

(b) ℓ_2-regularized *multi-class logistic regression classifier*: Now each incident is represented as a 41-dimensional vector $\mathbf{x}_i$ whose elements are either 0 or 1 (viewed as real numbers). Write a MATLAB script to implement the gradient descent algorithm to learn the parameters $\theta := \{\mathbf{w}_1, \ldots, \mathbf{w}_m\}$ of an ℓ_2-regularized multi-class logistic regression. Recall that the negative log-likelihood function of logistic regression for m classes and its gradients are given by:

$$\text{NLL}(\theta) = \sum_{j=1}^{n} \ln\left(\sum_{k=1}^{m} e^{\mathbf{w}_k^\top \mathbf{x}_j}\right) - \sum_{k=1}^{m} \mathbf{w}_k^\top \left(\sum_{j=1}^{n} 1(y_j = k)\mathbf{x}_j\right)$$

$$\nabla_{\mathbf{w}_y} \text{NLL}(\theta) = \sum_{j=1}^{n} \left(\frac{e^{\mathbf{w}_y^\top \mathbf{x}_j}}{\sum_{k=1}^{m} e^{\mathbf{w}_k^\top \mathbf{x}_j}} - 1(y_j = y) \right) \mathbf{x}_j, \quad y = 1, \ldots, m.$$

The ℓ_2-regularized objective function to be minimized and its gradients are given by:

$$f(\theta) := \text{NLL}(\theta) + \frac{\lambda}{2} \sum_{k=1}^{m} \|\mathbf{w}_k\|_2^2, \quad \lambda > 0,$$
$$\nabla_{\mathbf{w}_y} f(\theta) = \nabla_{\mathbf{w}_y} \text{NLL}(\theta) + \lambda \mathbf{w}_y, \quad y = 1, \ldots, m.$$

The pseudocode of the gradient descent algorithm appears below.

Initialize: $\theta_0 := \{\mathbf{w}_1^{(0)}, \ldots, \mathbf{w}_m^{(0)}\}$
For $t = 1, 2, \ldots, t_{\max}$, do:
  Evaluate gradients: $\nabla_{\mathbf{w}_k} f(\theta_t), k = 1, \ldots, m$
  Update weights: $\mathbf{w}_k^{(t+1)} = \mathbf{w}_k^{(t)} - \eta_t \nabla_{\mathbf{w}_y} f(\theta_t), k = 1, \ldots, m$
Endfor

Choose a fixed step size η (recommended: 10^{-5}) and initialize all $\mathbf{w}_k$'s to be zero. We recommend using $\lambda = 1000$. Use the first 60% of data samples in `data_SFcrime_train.mat` as your training set and the remaining 40% as the "test" set.

(i) **Plot** the value of the objective function $f(\theta_t)$ (y-axis) against the number of iterations t (x-axis) for $t = 1, 2, \ldots, t_{\max} = 1000$.

(ii) **Plot** the test *logloss* and the test CCR versus the number of iterations respectively. The *logloss* on the "test" set is defined as (see notes on Classification Performance Metrics for motivation and details):

$$\text{logloss} = -\frac{1}{n_{\text{test}}} \sum_{j=1}^{n_{\text{test}}} \log p(y_j | \mathbf{x}_j, \theta)$$

where $\mathbf{x}_j$ is the test data point and y_j is its ground truth label, and $p(y|\mathbf{x}, \theta) = \dfrac{e^{\mathbf{w}_y^\top \mathbf{x}}}{\sum_{k=1}^{m} e^{\mathbf{w}_k^\top \mathbf{x}}}$. In order to avoid taking the logarithm of extremely small numbers, any value $p(y_j|\mathbf{x}_j, \theta)$ that is smaller than 10^{-10} should be treated as 10^{-10}.

(c) (**Optional Bonus**) **Cross-validate** the regularization parameter λ using all the data in `data_SFcrime_train.mat`. **Report** the best value of λ. **Predict** the labels for the **real** test data points in `data_SFcrime_test.mat` (the ground truth labels for the real test data have been withheld by the competition). **Submit** your results to the original competition website and find out and **report** the real test error from the website. Report the strategy that you used and your results on the leader board of the competition. **Attach** screen-shots of your submission to the website and the leaderboard.