Homework II

Due date: Feb. 19, 2019 (Before the Class)

Problem 1: Least Absolute Deviation (15 points)

In class, we have assumed the following data generative model

$$y = f(\mathbf{x}) + \epsilon$$

where ϵ follows a standard gaussian distribution, i.e., $\epsilon \sim \mathcal{N}(\epsilon|0, 1)$. Assume a linear model for $f(\mathbf{x}) = \mathbf{w}^\top \mathbf{x}$. We now modify the data generative model by assuming that ϵ follows a Laplacian distribution whose probability density function is

$$p(\epsilon) = \frac{\lambda}{2} \exp(-\lambda|\epsilon|)$$

where λ is a positive constant. For more about Laplacian distribution please check the following wiki page `http://en.wikipedia.org/wiki/Laplace_distribution`.

Based on the above noise model about ϵ, derive the log-liklihood for the observed training data $\{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_n, y_n)\}$ and the objective function for computing the solution $\mathbf{w}$. Does the problem have a closed form solution like Least Square Regression?

Problem 2: Regression with Ambiguous Data (30 points)

In the regression model we talked about in class, we assume that for each training data point $\mathbf{x}_i$, its output value y_i is observed. However in some situations that we can not measure the exact value of y_i. Instead we only have information about if y_i is larger or less than some value z_i. More specifically, the training data is given as a triplet $(\mathbf{x}_i, z_i, b_i)$, where

- $\mathbf{x}_i$ is represented by a vector $\phi(\mathbf{x}_i) = (\phi_0(\mathbf{x}_i), \ldots, \phi_{M-1}(\mathbf{x}_i))^\top$
- $z_i \in \mathbb{R}$ is a scalar, $b_i \in \{0, 1\}$ is a binary variable indicating that if the true output y_i is larger than z_i ($b_i = 1$) or not $b_i = 0$

Develop a regression model for the ambiguous training data $(\mathbf{x}_i, z_i, b_i), i = 1, \ldots, n$.

Hint: Define a Gaussian noise model for y and derive a log-likelihood for the observed data. You can derive the objective function using the error function given below (note that there is no closed-form solution). The error function is defined as

$$\text{erf}(x) = \frac{1}{\sqrt{\pi}} \int_{-x}^{x} e^{-t^2} dt$$

It is known that

$$\frac{1}{\sqrt{2\pi}} \int_{-\infty}^{x} e^{-t^2/2} dt = \frac{1}{2}\left[1 + \text{erf}\left(\frac{x}{\sqrt{2}}\right)\right], \text{ and } \frac{1}{\sqrt{2\pi}} \int_{x}^{\infty} e^{-t^2/2} dt = \frac{1}{2}\left[1 - \text{erf}\left(\frac{x}{\sqrt{2}}\right)\right]$$

Problem 3: Regularization Penalizes Large Magnitudes of Parameters (15 points)

In class, we have learned that when increasing the regularization parameter λ in the regularized least square problem

$$\min_{\mathbf{w}} \frac{1}{2}\|\Phi\mathbf{w} - \mathbf{y}\|_2^2 + \frac{\lambda}{2}\|\mathbf{w}\|_2^2$$

where $\mathbf{y} = (y_1, \ldots, y_n) \in \mathbb{R}^n, \Phi^\top = (\phi(\mathbf{x}_1), \ldots, \phi(\mathbf{x}_n)) \in \mathbb{R}^{M\times n}$, the magnitude of the optimal solution will decrease. Let the optimal solution $\mathbf{w}_*$ be

$$\mathbf{w}_* = (\lambda I + \Phi^\top\Phi)^{-1}\Phi^\top\mathbf{y}$$

You are asked to show that the Euclidean norm of the optimal solution $\|\mathbf{w}_*\|_2$ will decrease as λ increases.

Hint: (1) use the result from the Problem 2 in homework 1. (2) for any vector $\mathbf{u} \in \mathbb{R}^d$ if $V^\top V = I$ where $V \in \mathbb{R}^{d\times d}$ then $\|V\mathbf{u}\|_2 = \|\mathbf{u}\|_2$

Problem 4: Ridge Regression and Lasso (40 points)

In this problem, you are asked to learn regression models using ridge regression and lasso. The data set that we are going to use is the Boston Housing data. The task is to predict the house price in suburbs of Boston. More information about the data can be found here `https://archive.ics.uci.edu/ml/datasets/Housing`.

For this assignment, we use the preprocessed data available here `http://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/regression.html`. There are two versions: the original version and the scaled version. The txt format and the Matlab format of the data are also given in the homework-2 package on ICON. For the txt format, the first column is the target output y, and the remaining columns are features in the form of (feature_index:feature_value). For the Matlab format, *data.X* is the data matrix of $n \times d$, and *data.y* is the target output of $n \times 1$. For question (1), (2), (3), please use the scaled version. For question 4, please use the original version. If we let $\mathbf{x} \in \mathbb{R}^d$ denote the feature vector, the prediction is given by $\mathbf{w}^\top\mathbf{x}$, where $\mathbf{w} \in \mathbb{R}^d$ contains the coefficients for all features. For this and next problem, we are not going to consider the intercept term [1]. Your are encouraged to try to include the intercept term in the model but it is not necessary to report the results. A simple way to add the intercept term is to augment the data by adding one additional feature with a constant value 1 and then learn the model $\mathbf{w}^\top\bar{\mathbf{x}}$ using the augmented feature vector $\bar{\mathbf{x}}$.

For solving ridge regression, you can compute an exact solution. For solving lasso, you can use the Matlab codes available here `http://www.cs.ubc.ca/~schmidtm/Software/lasso.html`. A python library is also available here `http://scikit-learn.org/stable/modules/generated/sklearn.linear_model.Lasso.html`. A Matlab script for loading data and for solving ridge regression and lasso is also given in the homework-2 package. If you use the Python module in sklearn, you can run lasso by

```
sklearn.linear_model.Lasso(alpha, fit_intercept=False)
```

[1] a model in the form $w_0 + \mathbf{w}^\top\mathbf{x}$, where w_0 is called the intercept term.

(1). Solution of Ridge Regression and Lasso: Set the value of the regularization parameter $\lambda = 1000$, compute the optimal solution for ridge regression and lasso. Report the optimal solutions for both ridge regression and lasso. (Note: If you use the Python module in sklearn, the value of alpha should be set to λ/n, where n is the number of training examples.)

(2). Training and testing error with different values of λ: (i) Take the first 400 examples as training data and remaining 106 examples as testing data. (ii) For each value of λ in [0, 0.01, 0.1, 1, 10, 100, 1000] run the ridge regression and lasso to obtain a model ($\mathbf{w}$) and then compute the root mean square error on both the training and the testing data of the obtained model. (iii) Plot the error curves for root mean square error on both the training data and the testing data vs different values of λ. You need to show the curves, and discuss your observations of the error curves, and report the best value of λ and the corresponding testing error.

(3). Cross-validation: Use the selected 400 examples as training, follow the 5-fold cross-validation procedure to select the best value of λ for both ridge regression and lasso. Then train the model on the 400 examples using the selected λ and compute the root mean square error on the testing data. Report the best λ and the testing error for both ridge regression and lasso.

(4). Repeat (3) using the original version of the data and compare the results with that obtained for (3).

Note: for a set of examples $(\mathbf{x}_i, y_i), i = 1, \ldots, n$, the root mean square error of a prediction function $f(\cdot)$ is computed by RMSE $= \sqrt{\sum_{i=1}^{n}(f(\mathbf{x}_i) - y_i)^2/n}$.

Problem 5: Ridge Regression and Lasso (Optional: 20 points)

Repeat the first three questions (1), (2), (3) as in Problem 4 on the provided GENE expression data (the description of the data can be found here `http://myweb.uiowa.edu/pbreheny/data/bcTCGA.html`). The feature matrix is saved in GENE_feature_matrix.txt and the output vector is saved in GENE_response.txt. A Matlab data format is also provided in the homework package named GENE_data.mat. Use the first 400 examples as training and the remaining as testing. (Thanks to Professor Patrick Breheny for providing the data and to Yaohui Zeng for preparing the data in txt format).

Note: since this data has a large number of features, the Matlab code that directly computes the closed-form solution for ridge regression could take a long time. You are allowed to explore some other libraries for ridge regression and Lasso, e.g., Tensorflow (Python), glmnet (R).