Machine Learning, Spring 2018
Homework 4

Due on 23:59 May 3, 2018
Send to *cs*282_01@163.*com*
with subject "Chinese name+student number+HW4"

1 Hoeffding Inequality

(1) Using binomial distribution, find the probability that a sample of 10 marbles will have $v \leq 0.1$ given that $\mu = 0.9$. (15 points)

(2) Using Hoeffding Inequality to bound the probability that a sample of 10 marbles will have $v \leq 0.1$ given that $\mu = 0.9$. (15 points)

Proof. (1) Let $Xi, i \in [1, 10]$ be the 10 samples of marbles. then $p(X_i = 1) = \mu = 0.9$, and

$$\begin{aligned} P(\nu \leq 0.1) &= P(\bar{X} \leq 0.1) \\ &= P(\sum_{i=1}^{10} X_i \leq 1) \\ &= (1-\mu)^{10} + \binom{10}{1}\mu(1-\mu)^9 \\ &= 9.1 \times 10^{-9} \end{aligned}$$

(2) $\nu \leq 0.1, \mu = 0.9$, then $\mu - \nu \geq 0.8$.

$$\begin{aligned} P(\nu \leq 0.1) &= P(\mu - \nu \geq 0.8) \\ &\leq e^{-2*0.8^2*10} \\ &= 2.76 \times 10^{-6} \end{aligned}$$

□

2 Bias-variance decomposition

For a random variable z, let $\bar{z}$ denote its mean. Suppose our observation is generated by the true function f as

$$y = f(x) + \epsilon,$$

where ϵ is normally distributed with zero mean and standard deviation σ. Training data set is $\mathcal{D} = \{x^{(i)}, y^{(i)}\}_{i=1}^{m}$, from which you learned your hypothesis function $h_{\mathcal{D}}$. The bias-variance tradeoff is an important aspect of data science projects based on machine learning.

Now for a new data point x^* out of $\mathcal{D}$, we want to investigate the expected error between the predicted value and the observation y^*, i.e.,

$$\mathbb{E}_{\mathcal{D},\epsilon}[(y^* - h(x^*))^2].$$

This error can be decomposed into three parts namely: **variance**, **bias**^{2}, and **noise**, where the expectation is taken over all possible training set $\mathcal{D}$. Here

$$\begin{aligned} \textbf{variance} &= \mathbb{E}_{\mathcal{D}}[(h(x^*) - \overline{h(x^*)})^2] \\ \textbf{bias}^2 &= [\overline{h(x^*)} - f(x^*)] \\ \textbf{noise} &= \sigma^2 \end{aligned}$$

with $\overline{h(x^*)} = \mathbb{E}_{\mathcal{D}} h(x^*)$.

The error **bias** is the amount by which the expected model prediction differs from the true value or target; while **variance** measures how inconsistent are the predictions from 1one another, over different training sets, not whether they are accurate or not. **Models that exhibit small variance and high bias underfit the truth target. Models that exhibit high variance and low bias overfit the truth target**.

The data scientists goal is to simultaneously reduce bias and variance as much as possible in order to obtain as accurate model as is feasible. However, there is a tradeoff to be made when selecting models of different flexibility or complexity and in selecting appropriate training sets to minimize these sources of error.

Question:

Show that

$$\mathbb{E}_{\mathcal{D},\epsilon}[(y^* - h(x^*))^2] = \textbf{variance} + \textbf{bias}^2 + \sigma^2.$$

Hints: first you may want to prove a lemma that for any random variable, it holds true that

$$\mathbb{E}[(z - \bar{z})^2] = \mathbb{E}[z^2] - \bar{z}^2,$$

so that

$$\mathbb{E}[z^2] = \mathbb{E}[(z - \bar{z})^2 + \bar{z}^2].$$

It follows that $(y^* - h(x^*))^2 = (y^*)^2 - 2h(x^*)y^* + h(x^*)^2$. Note that y^* and $h(x^*)$ are independent variables. The result follows from using the lemma twice. Also note $\mathbb{E}_{\epsilon}[(y^* - f(x^*))^2] = \sigma^2$. (20 points)

Proof.

$$\begin{aligned}
\mathbb{E}_{\mathcal{D},\epsilon}[(y^* - h(x^*))^2] =& \mathbb{E}_{\mathcal{D},\epsilon}[(y^* - \overline{h(x^*)} + \overline{h(x^*)} - h(x^*))^2] \\
=& \mathbb{E}_{\mathcal{D},\epsilon}[(y^* - \overline{h(x^*)})^2] + 2\mathbb{E}_{\mathcal{D},\epsilon}[(y^* - \overline{h(x^*)})(\overline{h(x^*)} - h(x^*))] + \mathbb{E}_{\mathcal{D},\epsilon}[\overline{h(x^*)} - h(x^*))^2] \\
=& \mathbb{E}_{\mathcal{D},\epsilon}[(y^* - f(x^*) + f(x^*) - \overline{h(x^*)})^2] + 2\mathbb{E}_{\mathcal{D},\epsilon}[(y^* - \overline{h(x^*)})]\mathbb{E}_{\mathcal{D},\epsilon}[(\overline{h(x^*)} - h(x^*))] \\
& + \textbf{variance} \\
& \text{As } \mathbb{E}_{\mathcal{D},\epsilon}[(\overline{h(x^*)} - h(x^*))] = \overline{h(x^*)} - \overline{h(x^*)} = 0 \\
=& \mathbb{E}_{\mathcal{D},\epsilon}[(y^* - f(x^*)^2] + 2\mathbb{E}_{\mathcal{D},\epsilon}[(y^* - f(x^*))(f(x^*) - \overline{h(x^*)})] + \mathbb{E}_{\mathcal{D},\epsilon}[(f(x^*) - \overline{h(x^*)})^2] \\
& + \textbf{variance} \\
& \text{As } \mathbb{E}_{\mathcal{D},\epsilon}[y^* - f(x^*)] = \mathbb{E}_{\mathcal{D},\epsilon}[\varepsilon] = 0, \mathbb{E}_{\mathcal{D},\epsilon}[f(x^*) - \overline{h(x^*)}] = f(x^*) - \overline{h(x^*)} \\
=& \mathbb{E}_{\mathcal{D},\epsilon}[\varepsilon^2] + [f(x^*) - \overline{h(x^*)}]^2 + \textbf{variance} \\
=& \textbf{variance} + \textbf{bias}^2 + \sigma^2
\end{aligned}$$

□

3 Cross Validation And L2 Regularization

Given the training data set "crime-train.txt" and the test data set "crime-test.txt", you can read them in the files with in Python:

```
import pandas as pd
df_train = pd.read_table("crime-train.txt")
df_test = pd.read_table("crime-test.txt")
```

This stores the data as Pandas DataFrame objects. DataFrames are similar to Numpy arrays but more flexible; unlike Numpy arrays, they store row and column indices along with the values of the data. Each column of a DataFrame can also, in principle, store data of a different type. For this assignment, however, all data are floats. Here are a few commands that will get you working with Pandas for this assignment:

```
df.head()                    # Print the first few lines of DataFrame df.
df.index                     # Get the row indices for df.
df.columns                   # Get the column indices.
df[''foo'']                  # Return the column named ''foo''.
df.drop(''foo'', axis = 1)   # Return all columns except ''foo''.
df.values                    # Return the values as a Numpy array.
df[''foo''].values           # Grab column foo and convert to Numpy array.
df.iloc[:3,:3]               # Use numerical indices (like Numpy) to get 3 rows and cols.
```

The data consist of local crime statistics for 1,994 US communities. The response y is the crime rate. The name of the response variable is `ViolentCrimesPerPop`, and it is held in the first column of `df_train` and `df_test`. There are 95 features x_i. These features include possibly relevant variables such as the size of

the police force or the percentage of children that graduate high school. The data have been split for you into a training and test set with 1,595 and 399 entries, respectively[1].

We'd like to use the training set to fit a model which can predict the crime rate in new communities, and evaluate model performance on the test set. As there are a considerable number of input variables, overfitting is a serious issue. In order to avoid this, implement the L2 regularization.

The main goal of this homework is to give you some experience using L2 regularization as a method for variable selection and using 10-folder cross-validation as a technique to get an insight on how the model will generalize to an independent dataset.Your function should accept a scalar value of λ, a vector-valued response variable ($\mathbf{y}$), a matrix of input variables ($\mathbf{X}$), and an initial vector of weights ($\mathbf{w}_0$). It should output a vector of coefficient values ($\hat{\mathbf{w}}$).

In your analysis, include:

(1) *(points)* A plot of $\log(\lambda)$ against the squared error in the 10-folder splited training data. (15 points)

(2) *(points)* A plot of $\log(\lambda)$ against the squared error in the test data. (10 points)

(3) *(points)* A plot of λ against the number of small coefficients (you can set a threshold), and a brief commentary on the task of selecting λ. (15 points)

(4) *(points)* For the λ that gave the best test set performance, which variable had the largest (most positive) coefficient? What about the most negative? Discuss briefly. (10 points)

Solution:

1. Set λ between 10^{-4} and 10^{3}, then the optimal λ is 20.49 which minimizes the average square error.

[1]The features have been standardized to have mean 0 and variance 1.

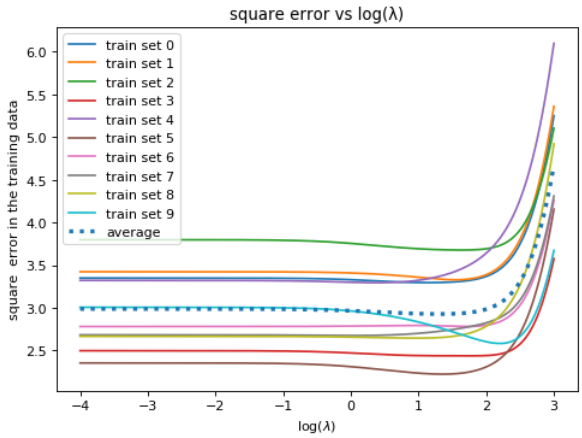

Figure 1: Square error vs log (λ)

2. The square error of the test data is 8.47

3. Set the threshold $5e-3$, then the figure of the number of small coefficients against the λ is as follow:

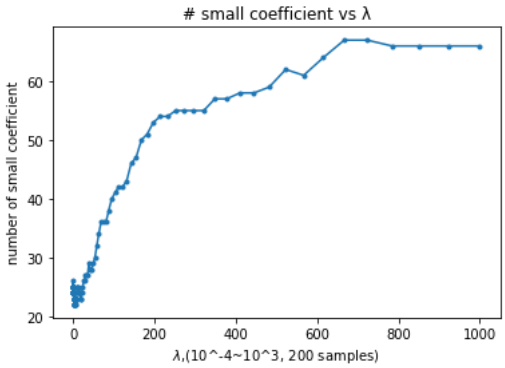

Figure 2: No. of small coefficients vs λ

4. Maximum coefficient is PctIlleg which value is about 0.061; Minimum coefficient is PctKids2Par which value is about -0.064. PctIlleg is percentage of kids born to never married; PctKids2Par is percentage of kids in family housing with two parent.
   The value of these coefficients mean that if fix other variables, then with every 1% increase in PctIlleg, the violent crime posibility will increase about 0.061%, and with every 1% increase in PctKids2Park, the violent crime possibility will decrease about 0.064%.