Assignment 1: k-nearest neighbours and linear regression

CS480/680 – Spring 2019
Submitted by: Kilby Baron, 20773955

May 25, 2019

1. **[30 pts]** Classification. Implement k-nearest neighbours using any programming language of your choice. Do not use any library such as scikit-learn that already has k-nearest neighbour or cross validation implemented. Implementing k-nearest neighbour and cross validation from scratch will be a good exercise to make sure that you fully understand those algorithms. Feel free to use general libraries for array and matrix operations such as numpy. Feel free to verify the correctness of your implementation with existing libraries such as scikit-learn.

   Download the dataset posted on the course web page. Classify each input x according to the most frequent class amongst its k nearest neighbours as measured by the Euclidean distance (L2-norm). Break ties at random. Determine the best number of neighbours k by 10-fold cross validation.

   What to hand in:

   - Your code for k-nearest neighbours and cross validation.
     See Q1.py
   - Find the best k by 10-fold cross validation. Draw a graph that shows the cross validation accuracy as k increases from 1 to 30.

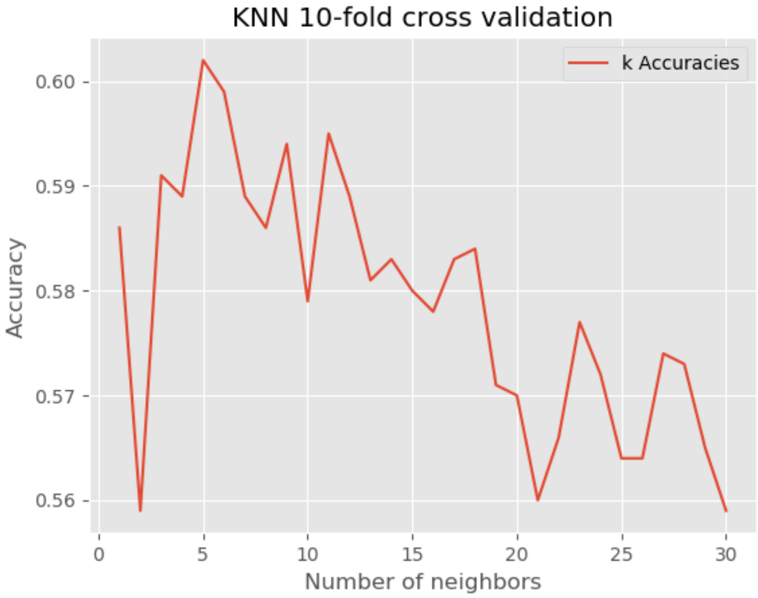

Figure 1: Cross validation accuracy as k increases from 1 to 30

- Report the best number of neighbours k and the accuracy on the test set of k-nearest neighbours with the best k.

  Best number of neighbours: 5

  Test set accuracy: 0.5454545454545454

2. **[30 pts]** Regression. Using any programming language, implement linear least square regression with the penalty term $0.5\lambda w^T w$. Do not use any library such as scikit-learn that already has linear regression or cross validation implemented. Implementing linear regression and cross validation from scratch will be a good exercise to make sure that you fully understand those algorithms. Feel free to use general libraries for array and matrix operations such as numpy. Feel free to verify the correctness of your implementation with existing libraries such as scikit-learn.

   Download the dataset posted on the course web page. The output space is continuous (i.e., $y \in \Re$). Determine the best λ by 10-fold cross validation.

   What to hand in:

   - Your code for linear regression and cross validation.

     See Q2.py

   - Find the best λ by 10-fold cross validation. Draw a graph that shows the cross validation accuracy as λ increases from 0 to 4 in increments of 0.1.

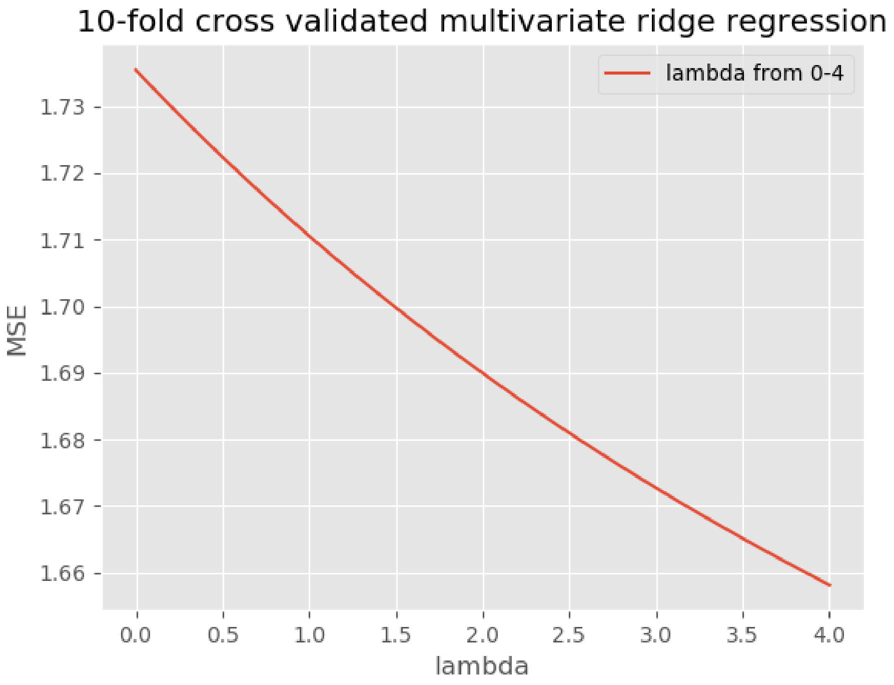

Figure 2: Cross validation accuracy as k increases from 1 to 30

   - Report the best λ and the accuracy of linear regression on the test set for this best λ.

   Best lambda: 4.0

   MSE of best lambda on test set: 1.7639

3. **[40 pts]** Theory. In class, we discussed several loss functions for linear regression. However all the loss functions that we discussed assume that the error contributed by each data point have the same importance. Consider a scenario where we would like to give more weight to some data points. Our goal is to fit the data points (x_n, y_n) in proportion to their weights r_n by minimizing the following objective:

$$L(w, b) = \sum_{n=1}^{m} r_n(y_n - wx_n + b)^2$$

where w and b are the model parameters, the training data pairs are (x_n, y_n). To simplify things, feel free to consider 1D data (i.e., x_n and w are scalars).

(a) **[20 pts]** Derive a closed-form expression for the estimates of w and b that minimize the objective. Show the steps along the way, not just the final estimates.

Find minimum by setting the derivative to 0:

$$\frac{\partial L}{\partial w} = \sum_{n=1}^{m} 2wr_n(y_n - wx_n + b) = 0$$

Solve for b:

$$\sum_{n=1}^{m} wr_n y_n - w^2 r_n x_n + wr_n b = 0$$

$$b = \sum_{n=1}^{m} \frac{wr_n y_n - w^2 r_n x_n}{wr_n}$$

$$b = \sum_{n=1}^{m} y_n - wx_n$$

Solve for w:

$$\sum_{n=1}^{m} wr_n(y_n - wx_n + b) = 0$$

$$\sum_{n=1}^{m} wr_n(y_n - wx_n + y_n - wx_n) = 0$$

$$\sum_{n=1}^{m} 2wr_n(y_n - wx_n) = 0$$

Assuming w is non-zero...

$$\sum_{n=1}^{m} (y_n - wx_n) = 0$$

$$w = \sum_{n=1}^{m} \frac{y_n}{x_n}$$

(b) **[20 pts]** Show that this objective is equivalent to the negative log-likelihood for linear regression where each data point may have a different Gaussian measurement noise. What is the variance of each measurement noise in this model?

The equation for linear regression with Gaussian noise is:

$$Pr(\mathbf{y}|\overline{\mathbf{X}}, \mathbf{w}, \sigma) = \mathbf{\Pi_{n=1}^{N}} \frac{\mathbf{1}}{\sqrt{\mathbf{2}\pi\sigma}} \mathbf{e}^{\frac{\mathbf{(y_n - w^T \overline{x}n)^2}}{\mathbf{2}\sigma^{\mathbf{2}}}}$$

The negative log likelihood for linear regression with Gaussian noise is:

$$\sum_{n} - \frac{(y_n - \mathbf{w^T \overline{x}_n})^{\mathbf{2}}}{2\sigma^2}$$

Subbing w and b derived above into $L(w, b)$ gives:

$$L(w, b) = \sum_{n=1}^{m} r_n (y_n - w x_n + b)^2$$

$$L(w, b) = \sum_{n=1}^{m} r_n (y_n - (y_n/x_n) x_n + y_n - w x_n)^2$$

$$L(w, b) = \sum_{n=1}^{m} r_n (y_n - w x_n)^2$$

This equation is equivalent to the negative log-likelihood for linear regression where r_n is equal to $\frac{1}{2\sigma^2}$

$$r_n = \frac{1}{2\sigma^2}$$
$$r_n 2\sigma^2 = 1$$
$$\sigma^2 = \frac{1}{2r_n}$$

Therefore the variance of measurement noise in this model is: $\frac{1}{2r_n}$