# Assignment 2: Ridge Regression and kNN

UVA CS 4501-03 :
Machine Learning (Spring 2018)

Out: Feb. 13 2018
Due: Feb. 25/ Sun midnight 11:59pm, 2018 @ Collab

**a** *The assignment should be submitted in the PDF format through Collob. If you prefer hand-writing QA parts of answers, please convert them (e.g., by scanning or using PhoneApps like officeLens) into PDF form.*

**b** *For questions and clarifications, please post on piazza.*

**c** *Policy on collaboration:*

*Homework should be done individually: each student must hand in their own answers. It is acceptable, however, for students to collaborate in figuring out answers and helping each other solve the problems. We will be assuming that, with the honor code, you will be taking the responsibility to make sure you personally understand the solution to any work arising from such collaboration.*

**d** *Policy on late homework: Homework is worth full credit at the midnight on the due date. Each student has three extension days to be used at his or her own discretion throughout the entire course. Your grades would be discounted by 15% per day when you use these 3 late days. You could use the 3 days in whatever combination you like. For example, all 3 days on 1 assignment (for a maximum grade of 55%) or 1 each day over 3 assignments (for a maximum grade of 85% on each). After you've used all 3 days, you cannot get credit for anything turned in late.*

## 1 KNN and Model Selection (K) (programming)

- Purpose 1: To implement a very simple classifier, k-nearest neighbors (KNN), from scratch.
- Purpose 2: To implement k-folds CV for classification.

This problem provides an introduction to classification using the KNN algorithm. When creating a classification algorithm, one has to make an assumption about the nature of the data. For example, if you are classifying cats vs dogs, you would probably not make the assumption that cats have the same colors as other cats. In our case, KNN makes the assumption that a data point has the same label as the most popular label of the k labeled data points that are closest to itself. The measurement of closeness that we will use is euclidean distance. We will test our implementation on *"Movie_Review_Data.txt"*

- Please use the "knn.py" template to guide your work. Please use the following instructions and follow the function names and descriptions in the template. Please use Numpy or other related package to implement the knn algorithm. Feel free to cross-check your implementation against sci-kit's. Other requirements or recommendations are the same as Homework1.
- 2.1 Please download the "knn.py" file and implement the read_csv method. The last column includes a 0 or 1 label.
- 2.2 Implement the fold method:

$$training, testing = fold(data, i, kfold)$$

Where data is the full data matrix, i is the iteration of cross validation, and kfold is the total number of folds. This method will be as simple as splitting the data matrix into training and testing sets.

- 2.3 Implement the classify method:

$$predictions = classify(training, testing, k)$$

Where training is the training set of data, testing is the testing set, and k is the number of data points to take into consideration when labeling an unlabeled data point. Note how the training data is part of the algorithm. In KNN, we label new data points based on the k points that are closest in our dataset. For each testing point, we find the k points in the training set that have the closest Euclidian distance to the testing point. The most popular label of the k closest points is the prediction.

- 2.4 Implement the calc_accuracy method:

$$acc = calc\_accuracy(predictions, labels)$$

Where predictions is the list of 0 or 1 predictions given from the classify method and labels is the true label for the testing points (the last column in the data matrix).

(Hint1: If your accuracy is below 50% look at the data, and consider how the order of the samples are dictated by the class)

- 2.5 Run the code with k = (3, 5, 7, 9, 11, 13). Report the accuracy and the best k. Discuss why some k values work better than others.
- A bar graph is recommended to show the change of accuracy with k. By using k as x-axis and accuracy as y-axis
- Att: we will not consider the speed in grading your kNN codes.
- Att: please remember to shuffle the whole data before performing the CV.
- Att: there are many ways to read in the reference datasets, e.g., our template reads in the whole file and put it into one numpy array. (But in HW1, our template actually read the file into two numpy array, one for Xval, the other for Yval. Both ways are correct.)

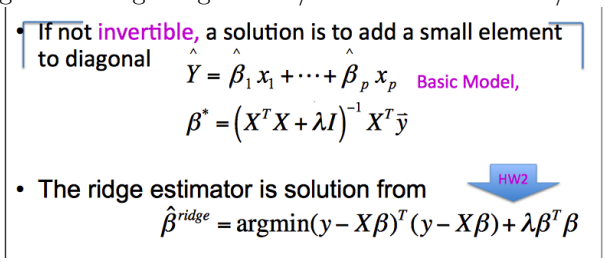
## 2 Ridge Regression (programming and QA)

- Purpose 1: To emphasize the importance of selecting the right model through k-folds Cross Validation (CV) when using supervised regression.
- Purpose 2: To show a real case in which linear regression learns badly and adding regularization is necessary.

This problem provides a case study in which just using a linear regression model for data fitting is not enough. Adding regularization like ridge estimator is necessary for certain cases.

- Here we assume $X_{n\times p}$ represents a data sample matrix which has $p$ features and $n$ samples. $Y_{n\times 1}$ includes target variable's value of $n$ samples. We use $\beta$ to represent the coefficient. (Just a different notation. We had used $\theta$ for representing coefficient before.)
- 1.1 Please provide the math derivation procedure for ridge regression (shown in Figure)

Figure 1: Ridge Regression / Solution Derivation / 1.1




(Hint1: provide a procedure similar to how linear regression gets the normal equation through minimizing its loss function. )
(Hint2: $\lambda|\beta|_2 = \lambda\beta^T\beta = \lambda\beta^T I\beta = \beta^T(\lambda I)\beta)$
(Hint3: Linear Algebra Handout Page 24, first two equations after the line "To recap,")

- 1.2 Suppose $X = \begin{bmatrix} 1 & 2 \\ 3 & 6 \\ 5 & 10 \end{bmatrix}$ and $Y = [1, 2, 3]^T$, could this problem be solved through linear regression? Please provide your reasons.
(Hint: just use the normal equation to explain)
- 1.3 If you have the prior knowledge that the coefficient $\beta$ should be **sparse**, which regularized linear regression method should be chosen to use ? (Hint: sparse vector)
- A data file named "RRdata.txt" is provided. For this data, you are expected to write programs to compare between linear regression and ridge regression.
- Please submit your python code as "ridgeRegression.py" . Please use the following instructions and use required function names. Please use Numpy or other related package to implement the ridge regression. Other requirements or recommendations are the same as Homework1.
- Notation: The format of each row in data file is $[1, x_1, x_2, y]$, where $x_1$, $x_2$ are two features and $y$ is the target value.
- 1.4 For "ridgeReregression.py",
  - Load the data file and assume the last column is the target value. You should use $xVal$ to represent the data sample matrix and $yVal$ to represent the target value vector.

- 1.4.1 The first function is to implement the ridge regression and return the coefficient $\beta$ with the hyperparameter $\lambda = 0$. (i.e. when $\lambda = 0$, it's just the standard linear regression). Please plot the data points and the learned plane [1]. Please submit the result into the writing part of this assignment. You are required to provide the following function (and module) for grading:

$$betaLR = ridgeRegression.ridgeRegress(xVal, yVal, lambdaV = 0)$$

- (Extra and not required Hint: In the state-of-the-art ridge regression implementations, the tools actually don't regularize the $\beta_0$. If you want to implement this strategy, you can estimate the un-regularized version $\beta_0$ through centering the input(i.e. $\hat{\beta}_0 = \frac{\sum y_i}{n}$) OR using the trick provided in the last EXTRA slide of our "ridge-regression" lecture.
- 1.4.2 The second function is to find the best $\lambda$ by using a $k = 10$ cross validation procedure (please feel free to try other $k$ like $k = 4$-fold). The function should be,

$$lambdaBest = ridgeRegression.cv(xVal, yVal)$$

- (Hint1: you should implement a function to split the data into ten folds; then loop over the folds; use one as test, the rest train )
- (Hint2: for each fold, on the train part, perform ridgeRegress to learn $\beta_k$; Then use this $\beta_k$ on all samples in the test fold to get predicted $\hat{y}$; Then calculate the error (difference) between true $y$ and $\hat{y}$, sum over all testing points in the current fold $k$. )
- 1.4.3 Please try all the $\lambda$ values from a set of values: $\{0.02, 0.04, 0.06, \ldots, 1\}$ (i.e. $\{0.02i | i \in 1, 2, \ldots, 50\}$). Pick the $\lambda$ achieving the best objective criterion from the 10-fold cross validation procedure. Our objective criterion is just the value of the loss function (i.e. $J(\theta)$ MSE in the slides) on each test fold. Please plot the $\lambda$ versus $J(\beta)$ graph (which is also called path of finding the best $\lambda$) and provide it into the writing. (ATT: the MSE is roughly in the range of $e - 2$. )
- Note : To constrain the randomness, please set seed to be 37. [2]
- Then run the ridge regression again by using the best $\lambda$ calculated from 1.4.2. Please include the result into writing.

$$betaRR = ridgeRegression.ridgeRegress(xVal, yVal, lambdaBest)$$

- Please plot the data points and the learned plane from best ridge regression. Please include the result into writing. [3].
- Att: there are many ways to read in the reference dataset, e.g., the ridge regression template reads in the file and put into two numpy array, one for Xval, the other for Yval. )

- 1.5 If assuming the true coefficient in problem 1.4 is $\beta = (3, 1, 1)^T$, could you compare and conclude whether linear regression or ridge regression performs better ? Explain why this happens based on the data we give.
  - (Hint: 1. Please implement a standard linear regression between $x_1$, $x_2$ and plot the $x_1$ versus $x_2$ graph;)
  - (Hint: 2. Guess the relationship between the two features and consider the problem 1.2.)
  - Please feel free to reuse your standRegress code from HW1

[1] `http://matplotlib.org/mpl_toolkits/mplot3d/tutorial.html#surface-plots`
[2] More about random in python, please see, `https://docs.python.org/2/library/random.html`
[3] `http://matplotlib.org/mpl_toolkits/mplot3d/tutorial.html#surface-plots`

# 3 Sample Exam Questions:

Each assignment covers a few sample exam questions to help you prepare for the midterm and the final. (Please do not bother by the information of points in some the exam questions.)

## Question 1. Short Answer

True or False? If true, explain why in at most two sentences. If false, explain why or give a brief counterexample in at most two sentences.

- (**True/False**). Ridge regression model increases the bias but reduces the variance comparing to the linear regression model.

- **(True or False?)** The error of a hypothesis measured over its training set provides a pessimistically biased estimate of the true error of the hypothesis.

- **(True or False?)** If you are given m data points, and use half for training and half for testing, the difference between training error and test error decreases as m increases.

- **(True or False?)** Overfitting is more likely when the set of training data is small.

- **(True or False?)** Overfitting is more likely when the hypothesis space is small.

- (**True/False**) When the tuning parameter $\lambda$ increases its value, the parameter $\beta$ in the ridge regression will not converge to zero vector, since Lasso enforces sparsity on $\beta$ (assuming no bias term here).

- (**True/False**). Ridge regression can fit the data well even if its feature variables have certain linearly dependent relationships among each other.

## Question 2. Bayes Rule (fake points)

(a) (4 points) I give you the following fact:

$$P(A|B) = 2/3$$

Do you have enough information to compute $P(B|A)$? If not, write "not enough info". If so, computer the value of $P(B|A)$.

(b) (5 points) Instead, I give you the following facts:

$$P(A|B) = 2/3$$
$$P(A|\sim B) = 1/3$$

Do you now have enough information to computer $P(B|A)$? If not, write "not enough info". If so, compute the value of $P(B|A)$.

(c) (5 points) Instead, I give you the following facts:

$$P(A|B) = 2/3$$
$$P(A|\sim B) = 1/3$$
$$P(B) = 1/3$$

Do you now have enough information to computer $P(B|A)$? If not, write "not enough info". If so, compute the value of $P(B|A)$.

(d) (5 points) Instead, I give you the following facts:

$$P(A|B) = 2/3$$
$$P(A|\sim B) = 1/3$$
$$P(B) = 1/3$$
$$P(A) = 4/9$$

Do you now have enough information to computer $P(B|A)$? If not, write "not enough info". If so, compute the value of $P(B|A)$.