**The report of Project-1**

张奕朗 16307130242

**1 Linear Regression and Nonlinear Bases**

**1.1 Adding a Bias Variable**

code files: leastSquaresBias.m and AddingABiasVariable.m

(1) Algorithm Skims

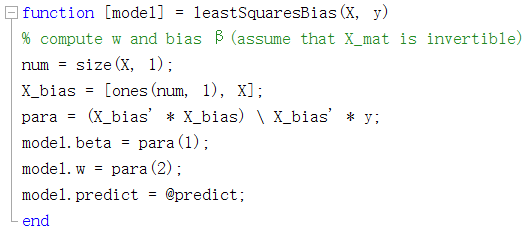
Least-squares linear regression minimizes the , and leads to a closed-form expression for the estimated value . The following, we consider adding a bias to improve our model to . In one-dimension case, . To minimize , let , we can get .

Transform them into matrix form to speed up computing :

, where , .

Let , we get as the least square solution.

(2) Critical Codes



In the codes above, X\_bias and para respectively represent for and .

(3) Experimental Analysis

In the improved model which has a bias term, we get updated training error = 3551.35 and test error = 3393.87. Compared to training error = 28122.82 and test error = 28298.97 in ‘example\_basis’, it’s a huge improvement.

Updated plot: The green line is the predicted for , and the blue points are real data in training set. We can see from the plot that when , , a huge deviation to , which manifest that adding a bias is reasonable to some degree.



(4) Discussion of Proposed Method

We can see from the plot that and are not strictly linear, but more likely a polynomial function.

**1.2 Polynomial Basis**

code files: leastSquaresBasis.m and PolynomialBasis.m

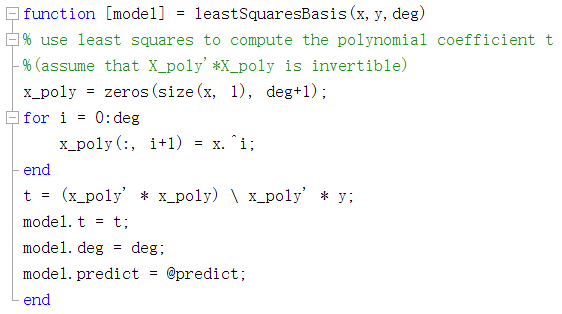
(1) Algorithm Skims

In the circumstance of polynomial .Similarly to the bias case, we have

, where , .

In order to minimize RSS (i.e. minimize the distance between and in higher dimensional space), .

(2) Critical Codes



In the codes above, I use a for-loop to generate the matrix , then compute the coefficient vector .

(3) Experimental Analysis

The results are as follows:

for deg = 0:

Training error = 15480.52, test error = 14390.76

for deg = 1:

Training error = 3551.35, test error = 3393.87

for deg = 2:

Training error = 2167.99, test error = 2480.73

for deg = 3:

Training error = 252.05, test error = 242.80

for deg = 4:

Training error = 251.46, test error = 242.13

for deg = 5:

Training error = 251.14, test error = 239.54

for deg = 6:

Training error = 248.58, test error = 246.01

for deg = 7:

Training error = 247.01, test error = 242.89

for deg = 8:

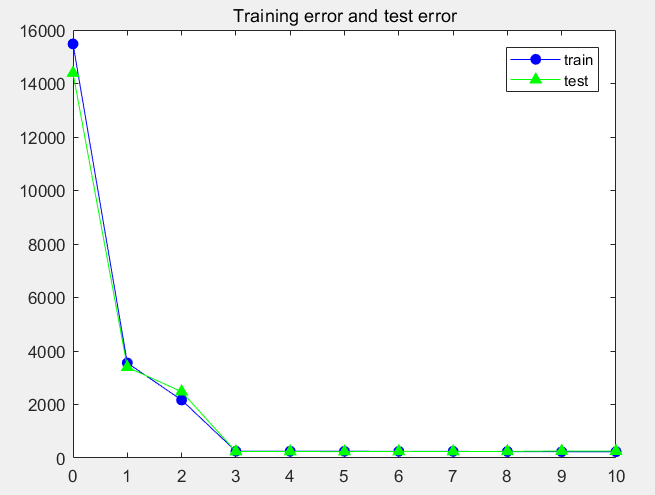
Training error = 241.31, test error = 245.97

for deg = 9:

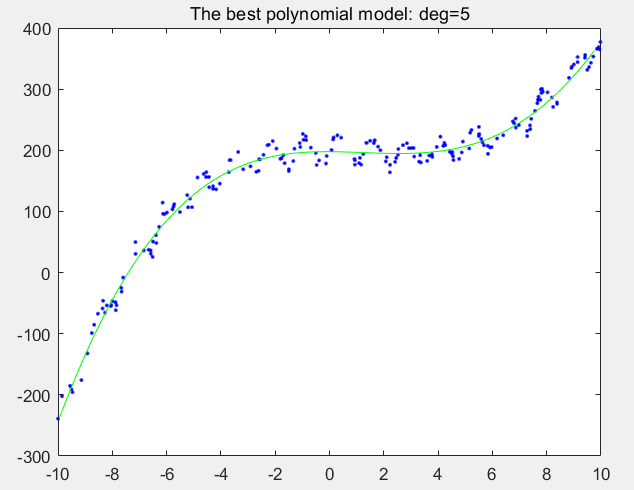
Training error = 235.76, test error = 259.30

for deg = 10:

Training error = 235.07, test error = 256.30



With the increasing of degree from 0 to 10, training error gets lower and lower, but test error gets lower until deg=5, then goes higher slightly, which manifests that overfitting occurred due to the increasing of complexity.

Choose the model with the lowest test error as the best model and plot it: The green line is the predicted for , and the blue points are real data in test set.

(4) Discussion of Proposed Method

Although the fitting is so good, there is still something need to notice: when the degree goes to 9 and 10, there is a warning by Matlab, ‘the matrix approaches singular values or scaling errors. The result may not be accurate. RCOND = 3.582888e-18.’ The reason for it is overflowing when the degree of gets too high, then underflowing when computing the pseudo-inverse of matrix .

A practical solution to this problem is standardization before computing, which makes .

**2 Regularization**

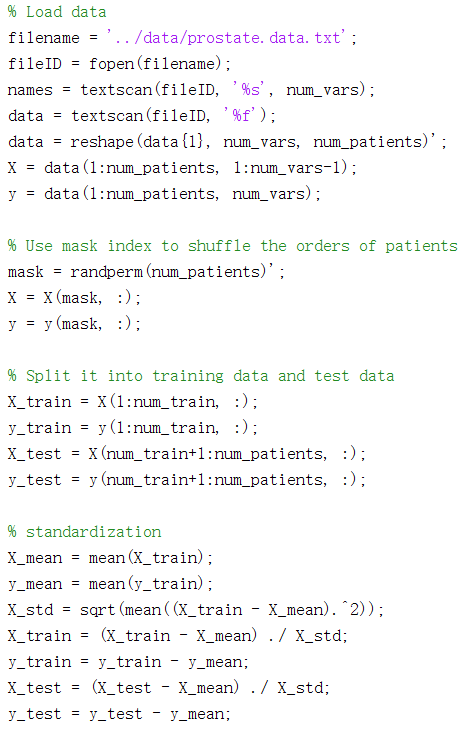
code files: RidgeRegression.m, CrossValidation.m and ridge.m

**2.2 Load and standardize the data**

Here are some constants in the code below, num\_vars, num\_patients and num\_train are respectively the number of variables (9), patients (97) and training data (50).

Use ‘textscan’ to load data from ‘prostate.data.txt’, then randomly shuffle the order and split them into training data (50 patients) and test data (47 patients).

Next, do standardization by the formula: , where is the empirical mean and is the empirical standard variance. And , where . Now, obeys the standard normal distribution and have zero mean.



There is a noticeable thing after doing standardization. and now both have zero mean, so , where is the bias term which is expected to be 0. (I’ll show the value of in the next section)

**2.3 Ridge regression**

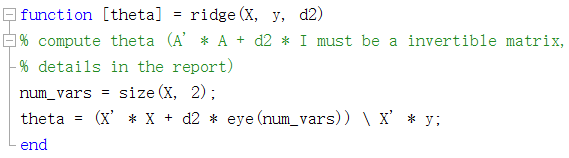
(1) Algorithm Skims

Ridge regression add a L-2 regularization term to RSS as the updated loss function , where is the regularization parameter, controlling the strength of L-2 regularization. Our aim is to minimize the loss function, so let , we get , where is an 8-by-8 unit matrix.

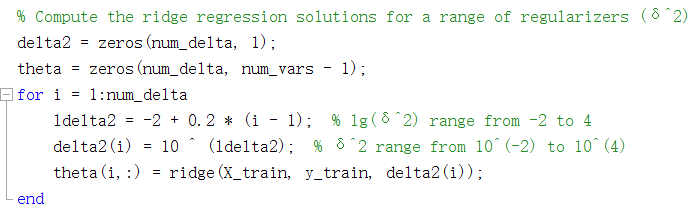
Notice that so is positive semidefinite. And we can infer that is a positive definite matrix which is invertible.

(2) Critical Codes

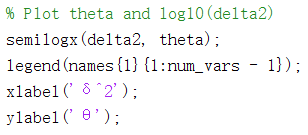
1. Function ridge:

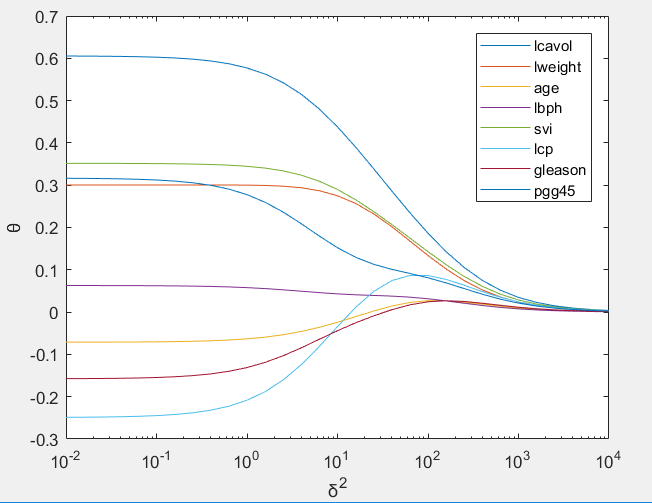


Then generate range from to , step by , and use the function above to compute corresponding parameter .

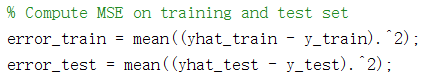


Plot and corresponding:

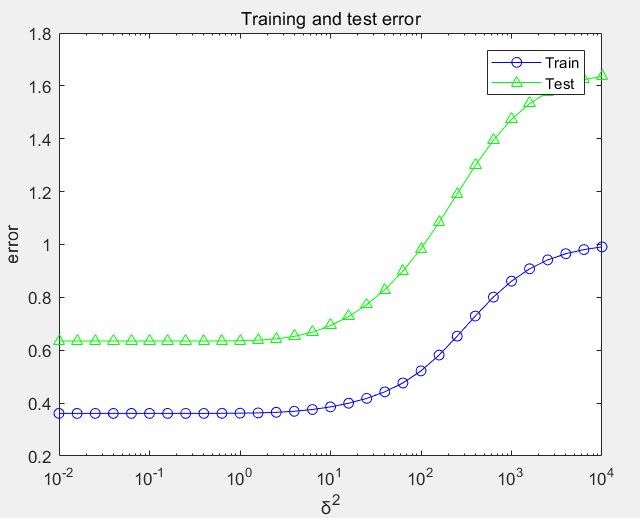




2. For each computed value of , compute MSE on training and test set. (See detailed values in variable error\_train and error\_test in RidgeRegression.m.)

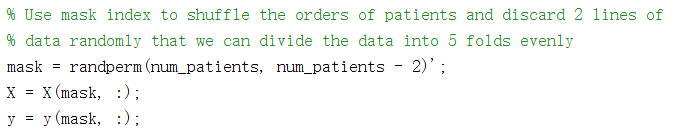


Plot them as follows:

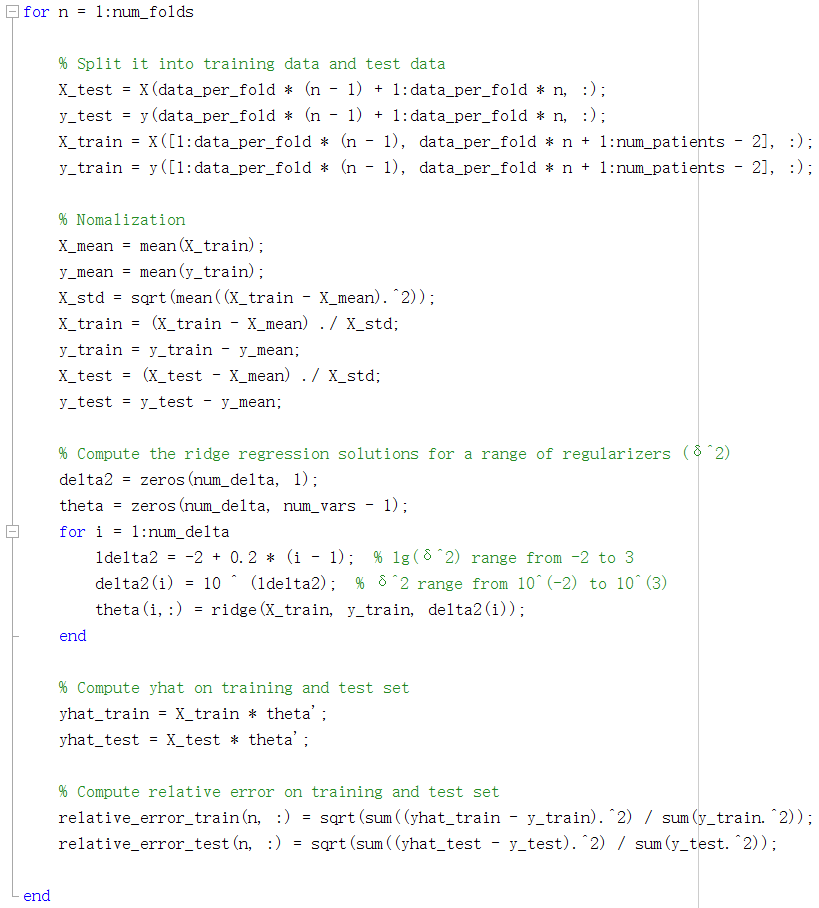


3. Use 5-folds cross validation to choose the value of with the lowest test error.

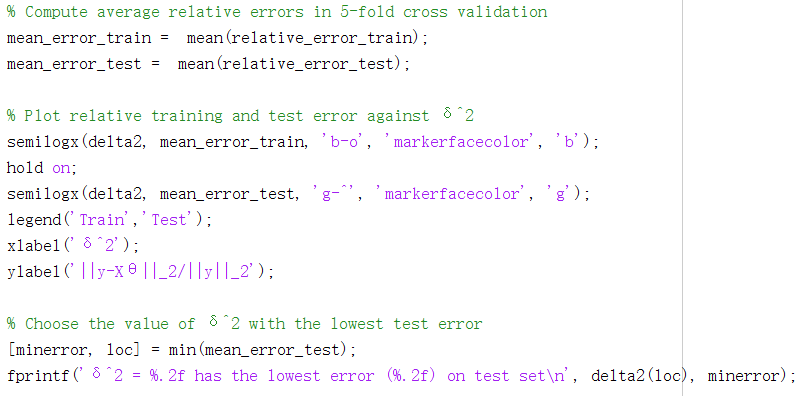
There are totally 97 patients’ data, in order to split them into 5 folds evenly, I abandoned 1 patients’ data randomly .

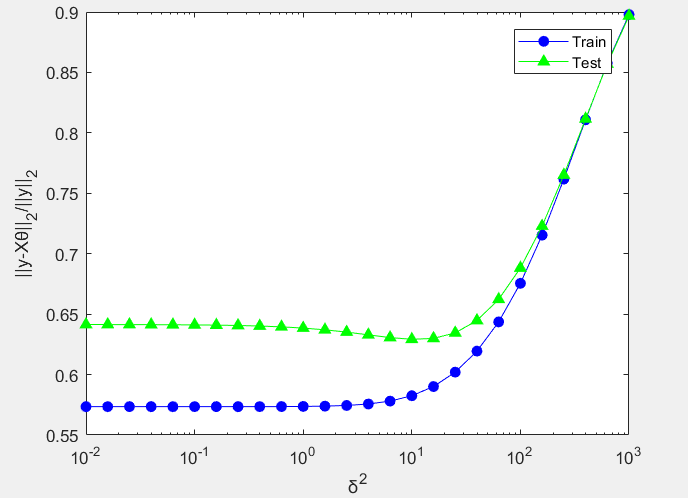


Then use a for-loop to do 5-folds cross validation, where ranges from to (from above we can know that is too high, so we shrink the range). Each time choose 1 fold as test set in proper order, the rest as training set.



Finally, compute and plot the average relative errors in 5 folds on training and test data, choosing the one with lowest test error:

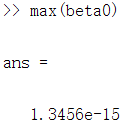




And we get (in this experiment) which has the lowest relative error (0.63) on test set.

(3) Experimental Analysis

① As I analyzed in last section, the expectation of the bias term is 0. I computed the values of for different values of in the file ‘RidgeRegression.m’ and store them in the variable ‘beta0’.



We can see that the maximum of is so small compared to the value of that is generated by random error and can be omitted. My analysis is confirmed by this phenomenon, so I omitted the bias term in my model.

② With the exponential increasing of , the relative error gets lower gradually, then goes higher quickly. It implies that proper regularization improved the model and excessive regularization forced the parameters become so small that didn’t fit the data.

③ It seems that the relative error is a little bit high, and the lowest one is about 0.63. This is because we use the square root of relative MSE as the error. Actually, the lowest relative MSE is , an acceptable error.

(4) Discussion of Proposed Method

The value of is a problem because the with the smallest test error varies from time to time, and it approximately located in (for 5-folds cross validation). I also tried 3-folds and 10-folds cross validation, the situation still exists. So, when we decide the value of , we may randomly choose a value from the range, or the magnitude of 10 may be a good choice.