Xie Zhou 912143385

ECS171 HW3

Problem1:

files: createTextFile.m problem1.m gene.txt;

run command: createTextFile , problem1

Answer:

NOTE: need to call createTextFile before calling problem1

NOTE: line 15 – line 40 took about 10 minutes to run. Because the other problems also call problem1, these lines were commented out. Its result was recorded in the following code. (It still runs properly)

---The regularization method I used was ridge.

1) It adds a constraint on weights in the original optimization problem. (sample size: n, number of features: m)

$$\text{Objective function: } \min \sum_{i=1}^{n} (y_i - wTx_i)^2$$

$$\text{Constraint: } \sum_{j=1}^{m} w\_j^2 \leq t$$

2) Then it can be rewritten as: (with no constraint field)

$$\text{Obj: } \min \sum_{\mathrm{i}=1}^{\mathrm{n}} (\mathrm{y_i} - \mathrm{wTx_i})^2 + \lambda \sum_{j=1}^{m} w\_j^2$$

3) Because it is a convex function, we can get the optimal solution by taking the derivative: (where I is a m-by-m identity matrix)

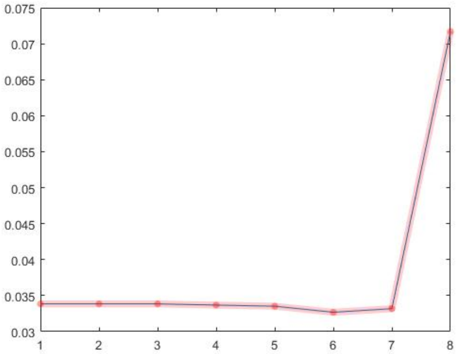
$$\text{Solution: w} = (wTw + \lambda \mathrm{I})^{-1} wTy$$

---The optimal λ value I chose was 0.01. I used 8 different λ (10e-5 – 10e2) values to run cross validation; recorded 8 corresponding average errors.

(line 15 – line 40)

1) error = [0.0338 0.0338 0.0338 0.0337 0.0335 0.0372 0.0332 0.0717]

2) So I picked λ = 0.01. Around 0.01, the change in error started to smooth out.



---Number of features: 2612
---10-fold cross validation generalization error : 0.0337

Problem2:
files: problem2.m gene.txt;
run command: problem2

Answers:
---Bootstrapping method:
1) sample size = 194, total number of samples = 10000.
2) For each sample, the 194 observations were collected from the original data (rows) randomly. They were picked with replacement, which means there could be duplicates of rows. Mean of 194 observations were taken, so in total there were 10000 sample means.
3) I sorted the collected means. To get the 95% confidence interval, take the 2.5% and 97.5% percentile.
4) So the 95% confidence interval is [0.3623, 0.4269]

Problem3:
files: problem3.m gene.txt;
run command: problem3

Answers:
1) Take the mean of all features columns, and multiply the result by w.
Note: In problem1, input feature matrix was added one column of 1's to it. Then the calculation of w was done. Therefore, w vector includes the constant "b" as well.
2) prediction = 0.3592

Problem4:

files: problem4.m   gene.txt;
run command: problem4

Answers:

Note:

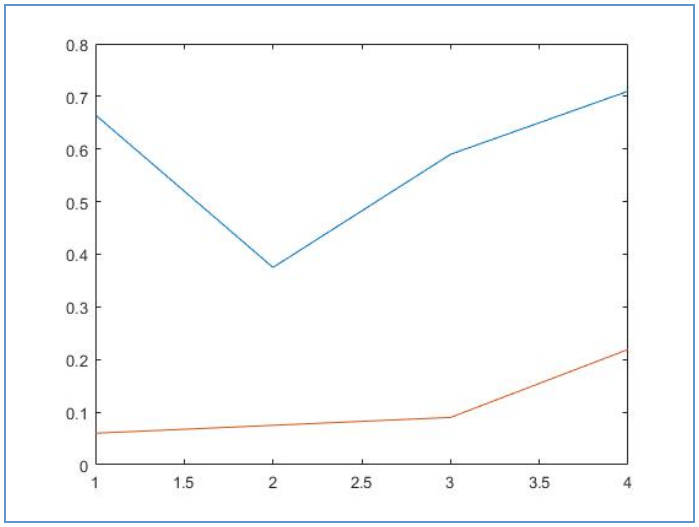
---I used a multiclass SVM classifier instead of a binary SVM. (fitcecoc svm toolbox instead of fitcsvm toolbox)

---There wasn't a sufficient way to set pairwise threshold for multiclass classifier. So instead of doing ROC, PR and AUC (they are for binary classification), I selected features with first and second highest posterior scores, and calculated their accuracies for the corresponding predictions.

---The number of features I used for all 4 classifiers was 2610, which was the result from problem 1 (ridge regularization).

---Performance analysis of 4 classifiers using10-fold cross validation method:

1) Choosing features with highest scores:
   Accuracy_strain = 0.6650
   Accuracy_medium = 0.3750
   Accuracy_environ = 0.5900
   Accuracy_gene = 0.7100
2) 2) Choosing features with second highest scores:
   Accuracy_strain = 0.0600
   Accuracy_medium = 0.0750
   Accuracy_environ = 0.0900
   Accuracy_gene = 0.2193



blue line: accuracy using highest score
red line: accuracy using second highest score

---gmTherefore, choosing the highest posterior score gives a much higher accuracy.

Problem5:
files: problem5.m   class_label.m   gene.txt;
run command: problem5

Answers:
---To create a composite classifier, I merged the two categories into one.

---There were 18 classes in "medium" and 8 classes in "environ". So in total, I had 18*8 = 144 labels for different classes in one category call "composite_y". (returned from class_label.m)

---Apply 10-fold cross validation on the composite_y classifier:

1) accuracy_composite = 0.2300
2) from problem 4, I calculated the combined accuracy of 2 individual classifiers by multiplying their accuracies (assuming independency):
   accuracy_medium+environ = 0.2169

abs(0.2300 – 0.2213)/0.2213 = 3.93%

---Therefore, there is a 3.93% increase in accuracy if composite classifier is used(not significant).

Problem6:
files: problem6.m   gene.txt;
run command: problem6

Answers:
1) To get the top 3 principal components:
   [u,s,v] = svd(x) (singular vector decomposition of original feature matrix X)
   s – singular values from top-left to bottom-right, descending order,
   v – corresponding right singular vectors
   pc1 = x*v(:,1);
   pc2 = x*v(:,2);
   pc3 = x*v(:,3);
   X = [pc1 pc2 pc3]

2) Train the four classifiers with the same y values (4 categories) and the new feature matrix "X".

3) 10-fold cross validation result:
   Accuracy_strain = 0.7300   (w/o PC = 0.6650) +9.77%
   Accuracy_medium = 0.2700   (w/o PC =0.3750) -28.00%

Accuracy_environ = 0.5107 (w/o PC =0.5900) -13.44%
Accuracy_gene = 0.7959 (w/o PC =0.7100) +12.10%

There's a drop in medium accuracy and environ accuracy, and an increase in the other two. The PCs moderately retained/balanced the classification performance while the dimensionality was reduced.

Note: Using only top 2 PCs results higher accuracies for all 4 classifications.