Stat406 Spring 2010: homework 1

1 CV for picking K in an KNN classifier (Matlab)

In this question, you will learn how to use cross-validation (Section 1.4.2.1) to select K for a K-nearest neighbor classifier.

1. Modify the function `knnClassifyDemo` to compute the error rate on the training and test sets for

$$K \in \{1, 2, 5, 10, 20, 30, 40, 50, 60, 70, 80, 90, 100, 110, 120\} \quad (1)$$

   Plot the error rates vs K, as in Figure 1(a). Turn in your code and plots.

2. We can define the **degrees of freedom** (dof) of a KNN classifier as N/K, since we average each of the N labels over K neighbors, so the effective number of labels the model can "memorize" is about N/K. A larger dof means the model is more complex. Plot the error rates vs log dof, as in Figure 1(b). Turn in your code and plots.

3. What value of K would you choose, based on test set performance?

4. In real applications, we don't have access to a test set to choose K. Instead we will use 5-fold cross-validation to pick K. You can use the provided function `Kfold` to compute the indices for each fold. Use the following code fragment:

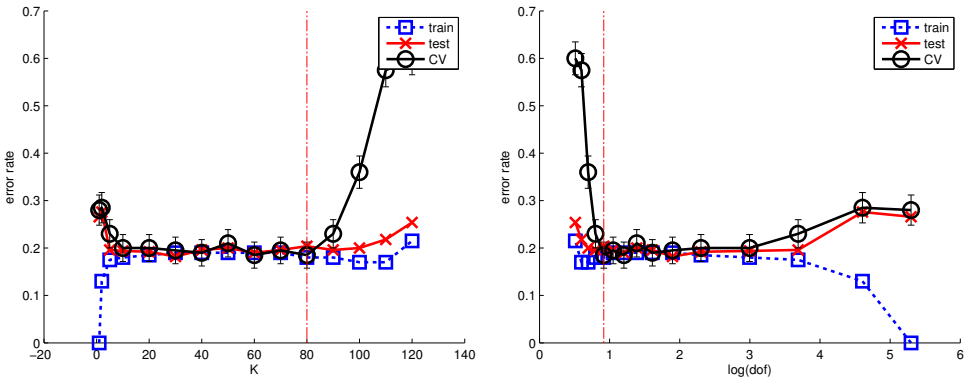
```
nfolds = 5;
[trainfolds, testfolds] = Kfold(Ntrain, nfolds);
for k=1:length(Ks)
  K = Ks(k);
  for i=1:nfolds
    XtrainFold = Xtrain(trainfolds{i},:);
    ytrainFold = ytrain(trainfolds{i});
    XtestFold = Xtrain(testfolds{i},:);
    ytestFold = ytrain(testfolds{i});
    [ypred] = knnClassify(XtrainFold, ytrainFold, XtestFold, K);
     .... insert your code here
  end
end
```

   Plot the CV estimate of the test error rate vs K and dof. The result should look like Figure 1 (you can ignore the error bars and the dotted vertical line). Turn in your code and plots.

5. What value of K would you choose, based on CV performance?

2 KNN classifier on shuffled MNIST data (Matlab)

Run `mnist1NNdemo` and verify that the misclassification rate (on the first 1000 test cases) of MNIST of a 1-NN classifier is 3.8%. (If you run it all on all 10,000 test cases, the error rate is 3.09%.) Modify the code so that you first randomly permute the features (columns of the training and test design matrices), as in `shuffledDigitsDemo`, and then apply the classifier. Verify that the error rate is not changed.



(a) (b)

Figure 1: Performance of a K-nearest neighbor classifier. (a) Error vs K. [Based on Figure 13.4 of [**?**]]. We see that the training error (blue square) is zero when $K = 1$, since we associate one label per training point. The test error goes up on the left due to overfitting, and goes up on the right due to underfitting. We also see that the shape of the CV curve (black circle) mirrors that of the test curve (red cross). Error bars represent 1 standard error of the mean. The red vertical line corresponds to the value of K chosen by the one-standard error rule (Section **??**). (b) Error vs ln of degrees of freedom. [Based on Figure 2.4 of [**?**].] Now the simpler models are on the left so the curves are reversed. Produced by Exercise 1.

3 Bayes rule for medical diagnosis

(Source: Koller)
After your yearly checkup, the doctor has bad news and good news. The bad news is that you tested positive for a serious disease, and that the test is 99% accurate (i.e., the probability of testing positive given that you have the disease is 0.99, as is the probability of tetsing negative given that you don't have the disease). The good news is that this is a rare disease, striking only one in 10,000 people. What are the chances that you actually have the disease? (Show your calculations as well as giving the final result.)

4 Prosecutor's fallacy

A crime has been committed in a large city and footprints are found at the scene of the crime. The guilty person matches the footprints. Out of the innocent people, 1% match the footprints by chance. A person is interviewed at random and his/ her footprints are found to match those at the crime scene. Determine the probability that the person is guilty, or explain why this is not possible. Hint: use Bayes rule.

5 The Monty Hall problem

(Source: Mackay)
On a game show, a contestant is told the rules as follows:

> There are three doors, labelled 1, 2, 3. A single prize has been hidden behind one of them. You get to select one door. Initially your chosen door will *not* be opened. Instead, the gameshow host will open one of the other two doors, and *he will do so in such a way as not to reveal the prize.* For example, if you first choose door 1, he will then open one of doors 2 and 3, and it is guaranteed that he will choose which one to open so that the prize will not be revealed.
>
> At this point, you will be given a fresh choice of door: you can either stick with your first choice, or you can switch to the other closed door. All the doors will then be opened and you will receive whatever is behind your final choice of door.

Imagine that the contestant chooses door 1 first; then the gameshow host opens door 3, revealing nothing behind the door, as promised. Should the contestant (a) stick with door 1, or (b) switch to door 2, or (c) does it make no difference? You may assume that initially, the prize is equally likely to be behind any of the 3 doors. Hint: use Bayes rule.