CS340 Fall 2007: Homework 1
Out 10 Sep, due 17 Sep

1 Bayes rule

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 diseas, 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.)

2 Bernoulli distributions

Let $X \in \{0, 1\}$ be a binary random variable (e.g., a coin toss). Suppose $p(X = 1) = \theta$. Then

$$p(x|\theta) = \text{Be}(X|\theta) = \theta^x(1-\theta)^{1-x} \tag{1}$$

is called a **Bernoulli** distribution. Prove the following facts:

$$E[X] = p(X = 1) = \theta \tag{2}$$

$$\text{Var}[X] = \theta(1-\theta) \tag{3}$$

3 Conditional independence

1. Let $H \in \{1, \ldots, K\}$ be a discrete random variable, and let e_1 and e_2 be the observed values of two other random variables E_1 and E_2. Suppose we wish to calculate the vector

$$\vec{P}(H|e_1, e_2) = (P(H = 1|e_1, e_2), \ldots, P(H = K|e_1, e_2))$$

   Which of the following sets of numbers are sufficient for the calculation?

   (a) $P(e_1, e_2)$, $P(H)$, $P(e_1|H)$, $P(e_2|H)$

   (b) $P(e_1, e_2)$, $P(H)$, $P(e_1, e_2|H)$

   (c) $P(e_1|H)$, $P(e_2|H)$, $P(H)$

2. Now suppose we now assume $E_1 \perp E_2|H$ (i.e., E_1 and E_2 are conditionally independent given H). Which of the above 3 sets are sufficent now?

Show your calculations as well as giving the final result. Hint: recall Bayes rule

$$P(H|\vec{e}) = \frac{P(\vec{e}|H)P(H)}{P(\vec{e})}$$

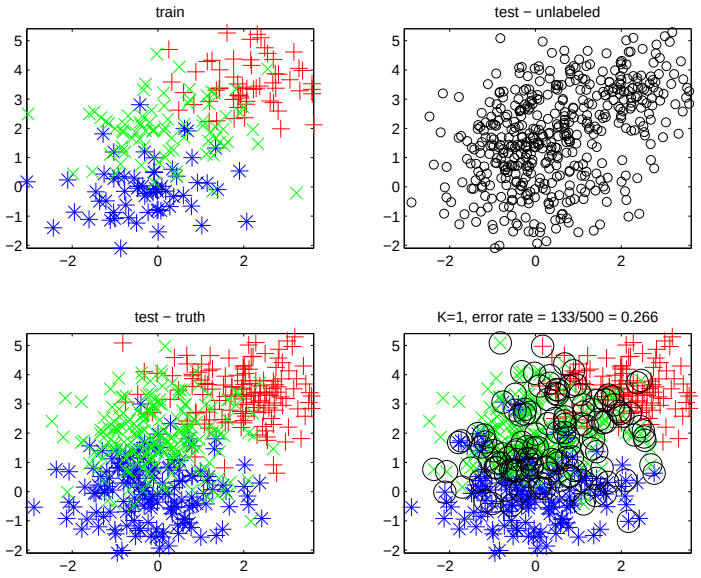

Figure 1: Data

4 kNN in Matlab

In this homework, we will learn how to plot data in Matlab, how to apply a kNN classifier, and how to use cross-validation to select k.

1. The file `knnClassify3CTrain.txt` contains 200 rows and 3 columns (separated by a space). The first 2 columns contain the input features, the last column contains the class label. Read this file using `dlmread` and create matrices `Xtrain` and `ytrain`. Similarly convert `knnClassify3CTest.txt` into `Xtest` and `ytest`. Turn in your code.

2. Plot the training data so that points in class 1 are red +s', and points in class 2 are blue *'s and points in class 3 are green x's. The result should look like Figure 1(top left). Turn in your code and plot.

3. You are provided a function

   ```
   [ypred] = knnClassify(Xtrain, ytrain, Xtest, K);
   ```

   that classifies each row of Xtest using the K-nearest neighbor algorithm. Apply this function to the test set using $K = 1$. Plot the test data with their predicted labels using the colors/ symbols above. Put a black circle around any points that are incorrectly classified. The result should look like Figure 1(bottom right). How many errors did your classifier make? Turn in your code and plot.

4. To visualize the prediction function $\hat{y} = f(x)$, we can apply it to a dense grid of test points x. The provided function `makeGrid2d`, which uses `meshgrid`, creates such a set of test points. Classify these test points and plot the result as follows:

   ```
   XtestGrid = makeGrid2d(Xtrain);
   ypredGrid = knnClassify(Xtrain, ytrain, XtestGrid, K);
   plotLabeledData(XtestGrid, ypredGrid) % you must implement this
   ```

   Do this for $K \in \{1, 5, 10\}$. The results should look like Figure 2. Turn in your plots. (We see that as K increases, the decision boundary tends towards a straight line, which is in fact optimal in this case (as we will see later), since the data was generated from a mixture of Gaussians.)

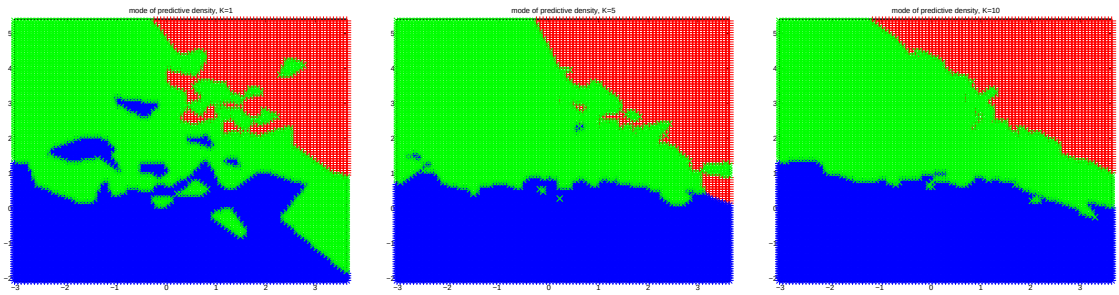

Figure 2: Predictive function for $K = 1$, $K = 5$ and $K = 10$

5. Now compute the error rate on the training and test sets for $K \in \{1, 2, \ldots, 20\}$. Plot the error rate vs the degrees of freedom, N/K. Use a log scale for the x-axis. The result should look like Figure 3(left). Also plot the training and test error vs K. The result should look like Figure 3(right). What is the best K? Turn in your code and plots.

6. 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);
Ks = [1:20];
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);
    errorRateFold(k,i) = ???
  end
end
```

   Plot the mean error rate vs K. Also plot the standard error, $se = \sigma/\sqrt{N}$, using the `errorbar` command. The result should look like Figure 3(right). The key point is that the CV curve has the same shape as the test curve. Turn in your code and plots.

References

[HTF01] T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning*. Springer, 2001.

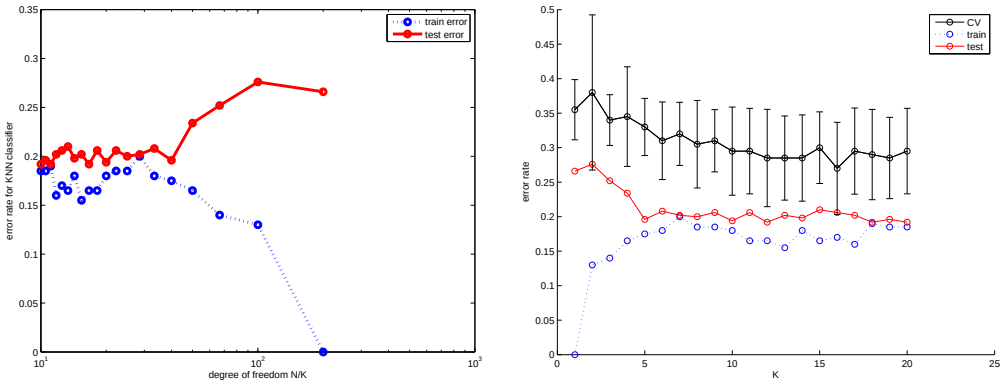

Figure 3: (a) Error vs dof. cf. Fig 2.4 of [HTF01]. (b) Error vs K. cf Fig 13.4 of [HTF01].