CSE 5522: HW 2 (Due 25 September 2015, 11:59 pm)

Problem 1 (80 points)

This problem will combine two different ideas: Bayesian networks and unsupervised clustering. We will build a classifier that first converts continuous features into discrete classes by a process called vector quantization, using an algorithm known as k-means clustering. The entire exercise is meant to be done programmatically. You must write your own k-means code – no borrowing, no libraries.

Given: a set of labeled data, containing training data points with 2-dimensional real values R and class labels C, and a set of testing data with similar labeling.

Procedure:

Step 1: we can develop a clustering function f: R->V, where V is one of a set of vectors resulting from K-means clustering. The function maps R to the closest vector V.

The k-means algorithm is relatively simple:

```
Initialization: choose $n$ vectors $V_1 \ldots V_n$ randomly
While not converged:
    For each datapoint $R_i$ find the closest vector $V_j$ and assign $R_i$ to $V_j$.
    Let $V_j$ <- average of all points assigned to $V_j$
    If no points are assigned to $V_j$ (or fewer than some threshold) randomly
        reinitialize $V_j$
```

Take the training points, *ignoring the current labels*, and perform k-means clustering with k=10 means.

Step 2: Once you have a set of vectors V, you can determine the closest vector V for each training data point R_i. From vectors representing the training set, compute a table $P(V|C)$ and compute $P(C)$. (You may also wish to compute $P(C|V)$ if you want to check your work.)

Step 3: On the test set, determine the class of each point by using $P(C|V) = \alpha P(V|C)P(C)$ -- note that I want you to do it this way rather than directly computing $P(C|V)$ because we will be using this idea later on in a future exercise. Report the average and standard deviation of the classification error rate over ten different runs of the k-means algorithm. Submit the working code.

Step 4: Sample the average/standard deviation of the classification error rate for k=2,5,6,8,12,15,20,50 (and other values if you wish). Qualitatively discuss the results of this experiment (1 paragraph).

For bonus questions: the code you submit should clearly be able to process the data in a way consistent with the different conditions outlined here.

Bonus 1 (10 points): The data were generated by sampling a set of Gaussians with random centers. Modify the data generator script (or write your own) to generate higher dimensional data (3d, 4d, 5d). Keep the number of classes and the Gaussian widths the same. Modify your program to repeat the experiment for k=10 – how do the results change as a function of dimensionality?

Bonus 2 (10 points): Modify the data generator script (or write your own) to increase the size of the Gaussian widths of the data generated for 2d data (so that they overlap more). Discuss how the results change as you increase widths.

Bonus 3 (10 points): Modify the data generator script (or write your own) so that the Gaussians are asymmetric – i.e. the Gaussian widths are different in different dimensions (use whatever dimensionality you wish, 2d is fine). Perform the experiment again. Think about the Euclidean distance metric – describe how the asymmetry affects the k-means process.

Problem 2 (20 points):

A particular football team will win 80% of the time with 1 quarterback playing during the game, 90% of the time with 2 quarterbacks playing and 70% of the time with 3 quarterbacks playing. The coach isn't saying ahead of time how many quarterbacks will play in the next game, although the prior probabilities on the decision are <.25,.5,.25> for 1,2,3 quarterbacks. Media scouting reports guessing the number of quarterbacks will be correct 60% of the time, and off by one in either direction a total of 40% of the time (equal probability of being plus or minus one). Note that you can't have 0 or 4 quarterbacks in the game.

a) Draw a Bayesian network describing the situation, including conditional probability tables.
b) Calculate the probability of P(Outcome=win | MediaReport=1QB).