Name:_____

EE435: Biometric Signal Processing Exam 1
Due Friday, 2/17/12 (Beginning of Class)

This is an individual exam. Do not communicate with anyone else: if you have questions, ask me. If you get stuck on a problem, you can ask me for help. In that case, I may determine a point cost for my help, based on the severity of the problem you are having. I will let you know how much the help would cost, then you will have the option to accept the help for a deduction in points, or you can go back and try again on your own.

You'll have this week's lab period to work on it, but it won't be due until Friday at the beginning of class. If turned in later, you will be penalized in points. Use any of your notes, the Digital Image Processing using MATLAB textbook (on the biometrics lab bookshelf), or online MATLAB help as needed.

There are also a few questions to answer, provide:

Hardcopy of:
Your answers to ALL questions in problems 1 and 2 (can be written on these sheets if there is room)
Two scatter plots for problem 2
Code for all parts of problem 2

Softcopy of:
The data.dat file from problem 2 where you have filled in the 2nd column

Problem	Points Possible	Points
1	**40**	
2	**60**	
Total:	**100**	

1. (40 pts total) Suppose you decide to use Bayesian decision theory in your senior design project, which is a pattern recognition system that uses signature biometrics. That is, identity is supposed to be based on how hard you press on a signature sensor as you sign your name. In your project, there are three midshipmen (Percy, Iggy and Cuthbert) who are enrolled in the system and who use it. The signature feature is measured only using three possible values: hard, medium and soft. You've taken some measurements of Percy, Iggy and Cuthbert over the course of the semester and determined some *a priori* knowledge about this situation. When reviewing the data you've collected about these three fellows and the lab, Percy enters the lab 35% of the time, Iggy enters 10% of the time and Cuthbert appears 55% of the time. You've also taken some measurements of their signature when they have entered the lab and have come up with the following information:

   Percy signs with hard pressure 20% of the time, signs with medium pressure 60% of the time, and with light pressure 20% of the time.

   Iggy signs with hard pressure 90% of the time, signs with medium pressure 8% of the time, and with light pressure 2% of the time.

   Cuthbert signs with hard pressure 20% of the time, signs with medium pressure 10% of the time, and with light pressure 70% of the time.

   a. (21 pts) Draw a probability tree included on the following page for this pattern recognition problem. Include probabilitistic expressions as well as numbers, so I can follow your tree's construction. Note: there is almost two pages of blank space that follows in this exam where you should include your work.

   b. (4 pts) What is the probability that it is Cuthbert entering the lab? Why?

   c. (5 pts) What is the probability that the person entering the lab is pressing hard on the signature sensor? Medium? Light? Show your work.

   d. (10 pts) Your system decides which person it is just by measuring the signature pressure. State your overall system decision rule by filling in the name of the person below (show work):

      If the person is pressing hard, it must be _____.

If the person is pressing medium, it must be _____.

If the person is pressing lightly, it must be _____..

2. (60 pts total) Download the file called *data.dat* from the course website. This file contains samples from 3 different classes. Each sample consists of 2 features.

   The matrix called *data* has 4 columns, laid out as follows:
   Column #1: Ground truth (class), either 1, 2 or 3
   Column #2: All zeros
   Column #3: Measurement of feature 1
   Column #4: Measurement of feature 2

   Use the load command to bring the data into MATLAB:

```
>> load data.dat -ascii
```

   a. (8 pts) Create a well-labeled scatter plot of this data, with feature 1 on the x-axis and feature 2 on the y-axis.

   b. (10 pts) From all this data, assume that only Class 2 is important to us right now. This means that for now, there are effectively two classes of interest: *Class 2*, and *NOT Class 2*. Determine the equation of a line that when applied to this data, can be used as a decision boundary to determine either *Class 2* or *NOT-Class 2* BUT with the following design constraint.

      Design constraint: using the given data, your decision boundary MUST allow <u>0 false accepts</u> in the given data and <u>minimize</u> the number of false rejects. Record your equation for the line below, and also <u>describe</u> how you determined that it met the design criteria.

      Equation:

   c. (6 pts) Using your equation from part b, state your decision rule for the *Class 2* or *NOT-Class 2* problem.

d. (6 pts) Write a MATLAB function called *classify2.m* that will incorporate your decision boundary and will decide whether the input feature vector is from class 2 or not.

Usage: `y = classify2(x)`

Input: `x` is the input feature vector [feature1 feature2].
Output: `y` = 1 if you decide input `x` is from class 2, and `y`=0 if not.
Error checking: input must be a vector with 2 values, or else y = [ ] and an error message is displayed.

e. (8 pts) Using the given data and your *classify2* function, write code to determine the number of false accepts and the number of false rejects for this *Class 2* or *NOT-Class 2* problem, as well as the FAR (%) and FRR (%) and record them:

FA=_____ FR=_____

FAR=_____ FRR=_____

f. (8 pts) Starting with your answer from part b-c, create a new linear decision boundary that will expand your decision rule to incorporate all three classes; i.e., the two linear boundaries will allow recognition of all three classes. Note: the linear boundary you already created will remain as is. Create another well-labeled scatter plot that includes all of your decision boundaries. Write your overall decision rule below.

Decision rule:

g. (6 pts) Write a MATLAB function called *classify.m* that will incorporate both of your decision boundaries and will decide whether the input feature vector is from class 1, 2 or 3.

Usage: `y = classify(x)`

Input: `x` is the input feature vector [feature1 feature2]
Output: `y` = 1 if you decide input `x` is from class 1, y = 2 if from class 2, y = 3 if from class 3.

Error checking: input must be a vector with 2 values, or else y = [ ] and an error message is displayed.

h. (8 pts) Write code that implements your 3-class decision rule, using your *classify* function, and computes the total number of errors and overall percent accuracy. Record your results below. Fill in the *data.dat* file's 2nd column with your classification of each sample as class 1, 2 or 3, and email that to me. To write out the file, use:

```
>> save data.dat data -ascii
```

Total # errors = _____

% Accuracy = _____ (show how you computed this)