EE435: Biometric Signal Processing

Project 2: Pattern Recognition I

Assigned: Tues 1/24/2012 Due: Tues 1/31/2012

I. Introduction

A typical pattern recognition problem is where we wish to be able to distinguish between *N* different classes of objects. In the salmon/sea bass example from class, there were two classes (*N*=2). To begin, we collect samples (i.e., images in our case) of each class and determine which features might be useful to tell one class from another. In the salmon/sea bass example from class, some possible features were: length, lightness, number/position of fins, etc. These features are measured and turned into numbers (to work on a computer), which become an element of a pattern vector. A pattern vector is an *M*-dimensional vector that contains the measurements of the *M* chosen features from a given object. Each object that is used in training or testing a system is represented by a pattern vector. After system design and testing is complete, a mathematical model is derived which can be thought of as a pattern vector that "best" represents a class; perhaps something like a "mean" pattern vector for each class.

During the system design and training phase, a training set of pattern vectors is used. The class each of these training pattern vectors belongs to is known *a priori*, so the accuracy of the system can be derived. To determine which features should be a part of the pattern vectors, it is important to get an idea of how much discriminatory information each feature contains about the classes individually (using a histogram, as you saw with the salmon/sea bass example) or in combinations (using a scatterplot). The system is then tested with a test set of pattern vectors, which are pattern vectors that the system has not seen before, but their ground truth (that is, the actual class they belong to) is known. These are used to test the performance of a system before it is placed on-line, to determine if the chosen number and types of features are "good enough". Design is an iterative process, so the training and testing can be repeated a number of times to determine the "best" system.

After the models for each class have been determined and the system is operational, new objects are processed to create a new pattern vector for which the class is unknown. These are compared in some fashion against the models for each class, and this comparison is used by the system to determine which class the new object belongs to.

In this project, you will be using the information in the "p.txt" file, which you can download from the course website under "Projects". This file contains 1500 samples (pattern vectors) from 3 classes (500 samples per class). There are two feature measurements per class. The overall goal of this project is to explore some of the concepts and visualizations involved in pattern recognition.

II. Histogram Analysis

Download the *p.txt* file from the course website. Use the load command to bring this data into MATLAB.

This will bring into your MATLAB workspace a variable called "p", which is a 1500 row x 3 column matrix. Consider each row a pattern vector. The columns are laid out as follows:

1st column: the true class of pattern (1 = class #1, 2 = class #2, 3 = class #3)—this is not really a feature, it is *ground truth*
2nd column: feature #1
3rd column: feature #2

1. For feature #1 (only), use the *histc* function to create a histogram of values of feature #1 for class 1 and for class 2 on the same plot, both in different colors. Type `>> help histc` to see how to use this function. Use *gtext* to label each curve on each plot as "Class #1" or "Class #2".

   When using the *histc* function, specify the edges of the bins for the histogram so that the width of the bins is 0.25, and so that the histogram spans from the minimum value to the maximum value of the feature. Label the axes. When you use the *histc* function, have it return the histogram values and then plot them using the *bar* function and the *hold on* command.

Question: If you only had to distinguish samples of class #1 and class #2 using feature 1, what value would you use as a threshold to differentiate between the two classes? Why? Comment on how separable class #1 and class #2 is using only feature 1.

2. On the same plot, add a histogram for class #3 (using feature #1 only) in a different color, and use *gtext* to label the histogram as "Class #3". **TURN IN THIS PLOT**.

   Question: Suppose you had to distinguish samples of all three classes using only feature 1. Comment on how separable these three classes are using just feature 1.

3. Repeat steps 1-2 above to create a plot of the histograms of all three classes using only feature #2. **TURN IN THIS PLOT**.

   Question: If you only could use one feature to distinguish these three classes, which one would you use and why?

III. Scatter Plot Analysis

A scatter plot is a two-dimensional (when considering only two features) or three-dimensional (for three features) plot that gives indication of the separability of the classes. We can only visualize a one, two or three-dimensional plot, so scatterplots are limited to three features for visual analysis. Pattern vectors with more than three elements (features) are typically used in a pattern recognition system, but we cannot visualize them. These higher order vectors, however, can be processed mathematically on a computer (more on this later in the course).

1. Create a two-dimensional scatterplot for the p.txt data. Use a single *plot* command in which you specify that class #1 patterns are blue squares, class #2 are red x's, and class #3 are black o's. See the help for the *plot* command for more details. Feature #1 should be along the horizontal axis, and feature #2 along the vertical axis. Be sure to include *xlabel* and *ylabel* commands to label the axes, and include a legend. It is very important that you use a legend for this plot, so samples of the different classes can be distinguished visually.

   Question: Would it be possible to draw decision boundaries such that any of the three classes are identified without any errors? Why or why not?

2. Using MATLAB, add the following parabolic-shaped decision boundary on top of the same plot (use the *hold* command to put it on the same plot). The equation for the parabolic boundary is:

   $f_2 = 0.25\ (f_1 - 5)^2 + 0.5$, where f_1 is a value of feature #1 and f_2 is the corresponding value of feature #2

   To do this, let the values of f_1 range from -5 to 15 with a step size of 0.01, then plug this f_1 vector into the above equation to determine the corresponding values of f_2, then plot f_1 vs f_2. **TURN IN THIS PLOT.**

   Question: This boundary is only good to distinguish one class from the other two classes…it cannot distinguish all three from each other. This means the decision is a binary decision, and results in "class x" or "not class x". Which class should this decision boundary be used for and why?

3. Use this decision boundary to determine the percent accuracy for this one class (call it class "x"). To do this, we turn the decision boundary equation into a decision rule:

   If $f_2 - \left[0.25\ (f_1 - 5)^2 + 0.5\right] \geq 0$, decide class x, where class x is your answer to question 2.

   If $f_2 - \left[0.25\ (f_1 - 5)^2 + 0.5\right] < 0$, decide NOT class x.

   Plug the feature values from the p.txt file into this equation and see if the output is ≥ 0. This will tell you what class this pattern recognition system would decide the features are from…if you compare it to the ground truth (i.e., the actual class given in the 1st column of the p.txt file), you can count how many you got wrong. The accuracy is then:

$$\%\ \text{Accuracy} = \frac{1500 - (\#\,\text{Errors})}{1500} \times 100\%$$

   Fill in these values:

   # Errors = _____

   % Accuracy = _____

   Question: In terms of accuracy, compare what you saw in Part II of this project (using one feature to recognize the three classes) vs what you saw in Part III (using two features)…which should provide better results?

   Question: Looking at the scatter plot with the parabolic decision boundary, what simple decision boundary could you add to the plot to help distinguish between all three classes? What would your new decision rule be?

For this project, include your code and all of the plots called for. Answer the questions asked.