CSCI-737 Pattern Recognition Assignment 1

Prasanth Ganesan
Doctoral Student, Biomedical Engineering,
Rochester Institute of Technology

February 8, 2016

Contents

1 Question 1 - Example of a recognition problem

Pattern recognition (PR) is an interesting field comprising of algorithms for recognizing patterns from a set of inputs. The field has a wide scope of application, primarily because of the generic nature of input to these algorithms [1]. Some common applications are text recognition, face recognition and various other tasks in signal and image processing. I would like to focus on an unique application based on my current research topic which is signal analysis of intra-atrial electrograms observed during a well-known cardiac arrrhythmia called atrial fibrillation (AF). When a person is experiencing AF, depending on the position of the catheter (voltage recording device), the activations in the electrograms recorded in different channels can form different patterns [2]. The ultimate task here is to classify the correct cycles of activations based on the observed patterns. A better understanding of the problem can be obtained from Figure 1 and Figure 2. Both the figures illustrate the results from a realistic simulation of a $2D$ space in the left atrium of a human heart during AF. Figure 1 show the patterns of simulated signals at various locations of the heart. Studies have shown that, exactly at the source of AF, the electrograms demonstrate a unique pattern which consists of sequential, temporal activations [3] (see "Pattern 4" in Figure 1). These are recordings from a 10-channel catheter – see the numbering on the left of the figure. For each channel, the x-axis represents time and y-axis represents amplitude (or voltage). Each pattern shows two cycles of activations (note that there are two peaks in each channel).

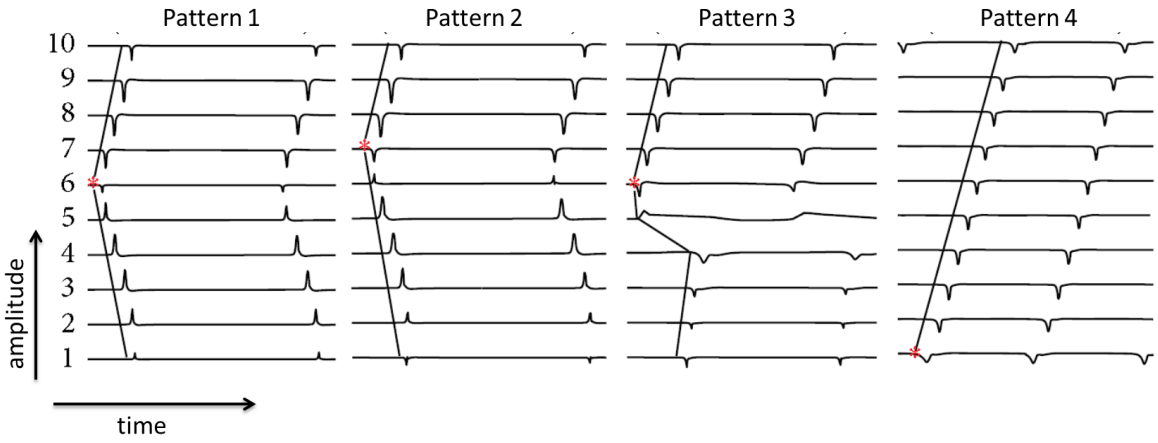

Figure 1: Overview of the Method – Figure shows the different patterns of simulated AF signals. x-axis represents time and y-axis represents amplitude. The numbers on the left indicate the recording channel. The thick black lines indicate the pattern of signals and the way in which the edges of the output graph would be connected. Figure reproduced from [2]

Before applying a PR method, the signals shown in the figure should be transformed to points in a $2D$ space, since the input to most algorithms are these generic point clouds. The time vs amplitude can be converted to normalized x-y scales. Separation of signals in the channels can be easily performed by providing an offset for each channel.

Now the task is three-fold:

- Identify the peak points (in my example, there are two peaks per channel) – This

is a classification problem

- Determine the the separation between all the peaks and store the points of interest – This is a segmentation problem

- Recognize the arrangement of the points of interest in the space, and hence find the relationship between the points, for example, recognizing the pattern in the locations of first peak in channel-1 (see Figure 1) and that in channel-2; this process is continued for all the peaks – This is a parsing problem.

From the above tasks, we can clearly realize the relationship between the three PR tasks itself. For classification, we need the arrangement of peaks; for segmentation, we need the different classes; and finally, for parsing, we need the segmented points of interest (which are essentially the peaks).

The input graphs for my example will have a vertex at every instant of sampled signal and the edges will be connected between all the vertices (in both dimensions). After applying PR, the output graph will essentially have vertices only at peaks, and the edges only connecting those vertices that belong to their respective cycle. For example, there are two cycles in Figure 1, hence, the resultant graph will be two separate graphs with vertices at peaks and edges following the pattern indicated in the figure. Obviously, all graphs here are undirected.

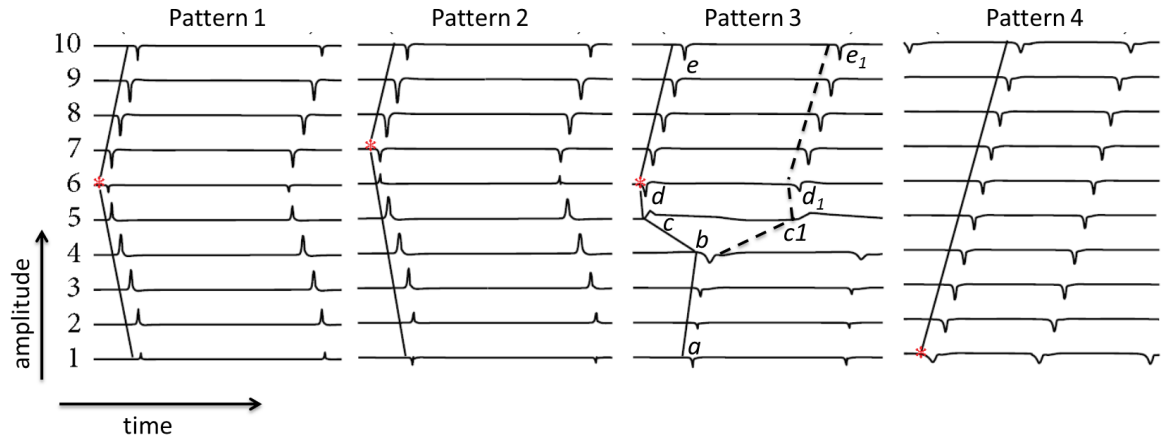

Figure 2: A specific PR task – Pattern-3 shows an interesting scenario where deciding on the classification cycles is relatively challenging. Figure reproduced from [2] and edited for this assignment

A challenging scenario in my example could be an activation pattern "Pattern 3" shown in Figure 2. The actual cycle is $a \implies b \implies c \implies d \implies e$, as indicated in the figure. However, by visual inspection one could get confused between a,b,c,d,e and a,b,c1,d1,e1, because of the larger space between channel-4 and channel-5. It would be even more challenging if a,b was a little more shifted to the right. I believe that we could make use of PR algorithms to classify such scenarios accurately, because these algorithms sometimes "see" what humans are not capable of.

2 Question 2 - Two well-known classification methods

Classification involves learning parameters from the training set and then dividing the space into regions of classes. Some basic algorithms are classifying using least squares method (assuming a linear model) and K-Nearest Neighbors method. Interestingly, these methods work pretty well most of the time, compared to more complex algorithms.

(a) Classification using Least Squares

The method assumes a linear model and finds the classes based on minimization of the sum of squared error. The problem was implemented in python 3.4. Figure 3 shows the input dataset (the orange and blue circles), the classified regions (green and yellow regions) and the classification line (black line).

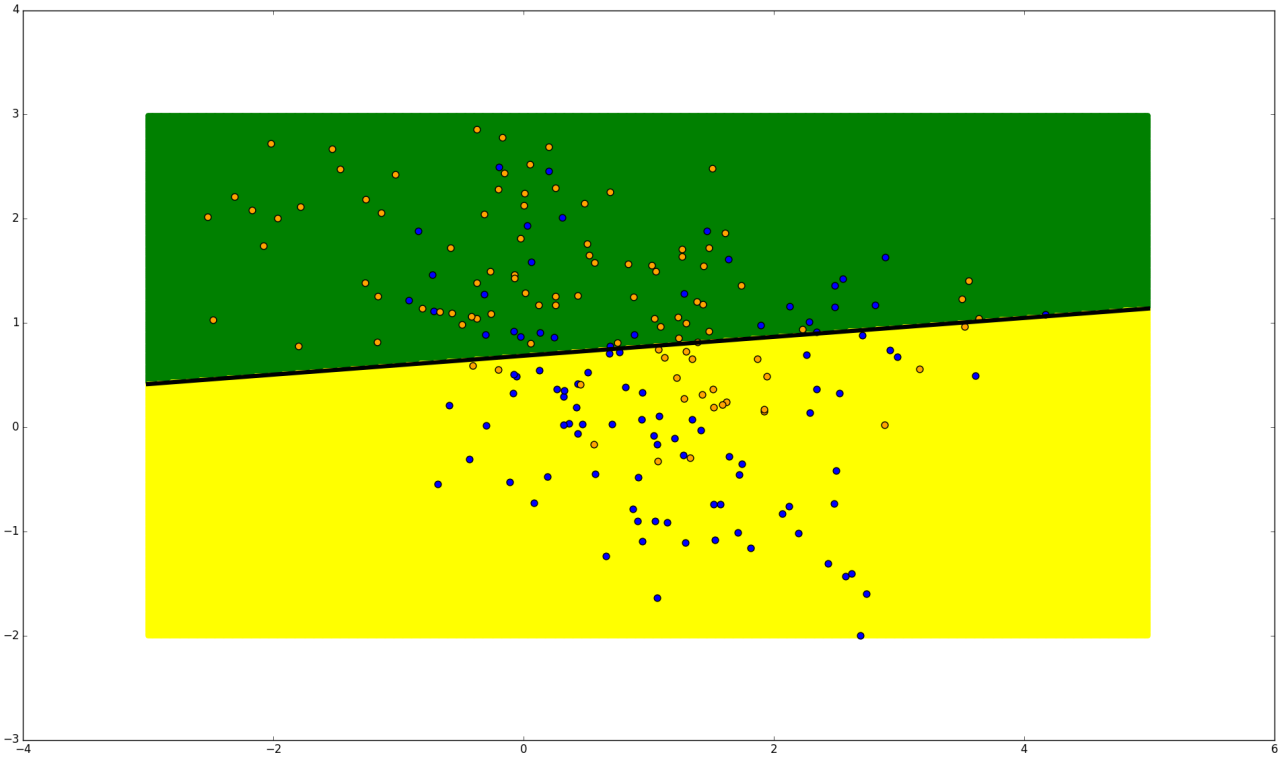

Figure 3: Classification using Least Squares – The orange and blue circles represents the input points (Blue = Class 0, Orange = Class 1), The green and yellow represents the regions (with same respective class labels) and the black line in between the regions represent the classification line obtained using this method

The classification rate using least squares was 73%.

(b) Classification using K-Nearest Neighbors

This method takes the most frequent class within a set of nearest neighbors defined by the user.

One Neighbor

Figure 4 shows the classification using $K = 1$ nearest neighbor. The problem was implemented in python 3.4. Figure 4 shows the input dataset (the orange and blue circles), the classified regions (green and yellow regions) and the classification boundary (black line).

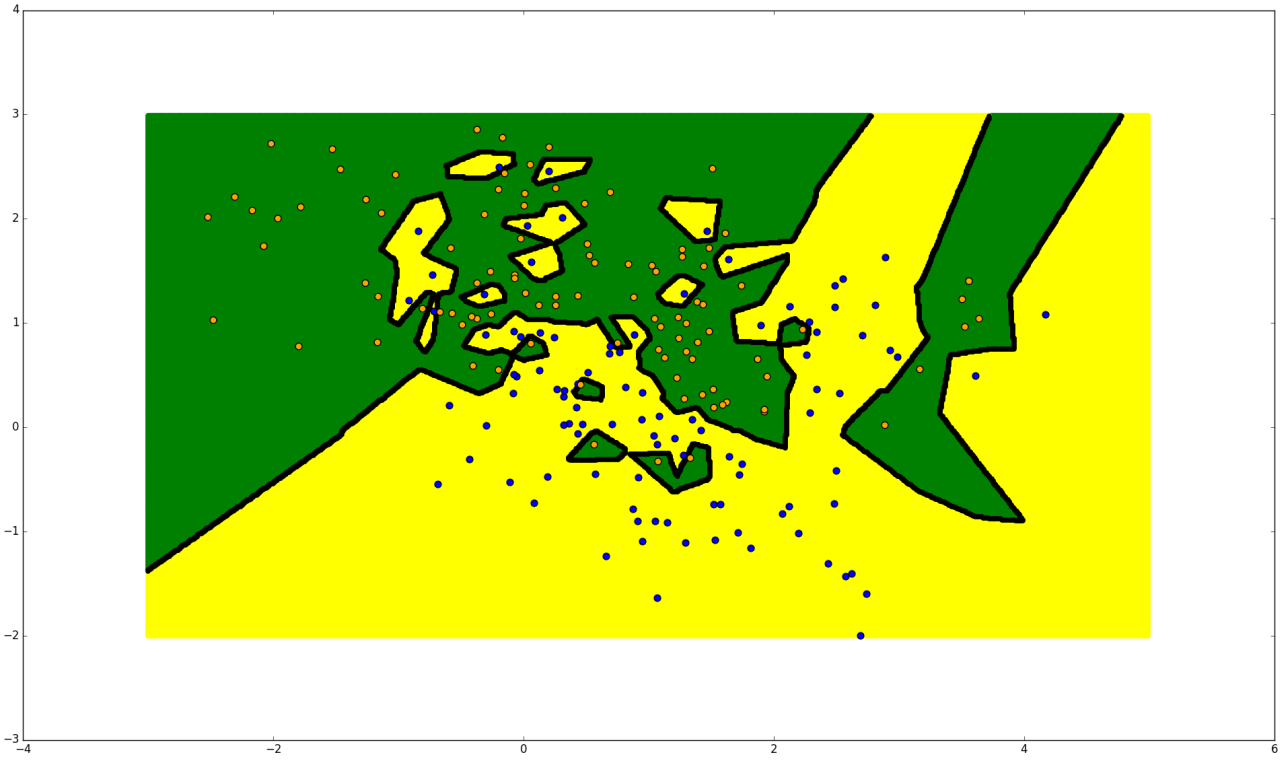

Figure 4: Classification using K-NN, $K = 1$ – The orange and blue circles represents the input points (Blue = Class 0, Orange = Class 1), The green and yellow represents the regions (with same respective class labels) and the black boundary in between the regions represent the classification boundary obtained using this method

The classification rate using 1-nearest-neighbor was 100%.

Fifteen Neighbors

Figure 5 shows the classification using $K = 15$ nearest neighbors. The problem was implemented in python 3.4. Figure 5 shows the input dataset (the orange and blue circles), the classified regions (green and yellow regions) and the classification boundary (black line).

The classification rate using 1-nearest-neighbor was 84.5%.

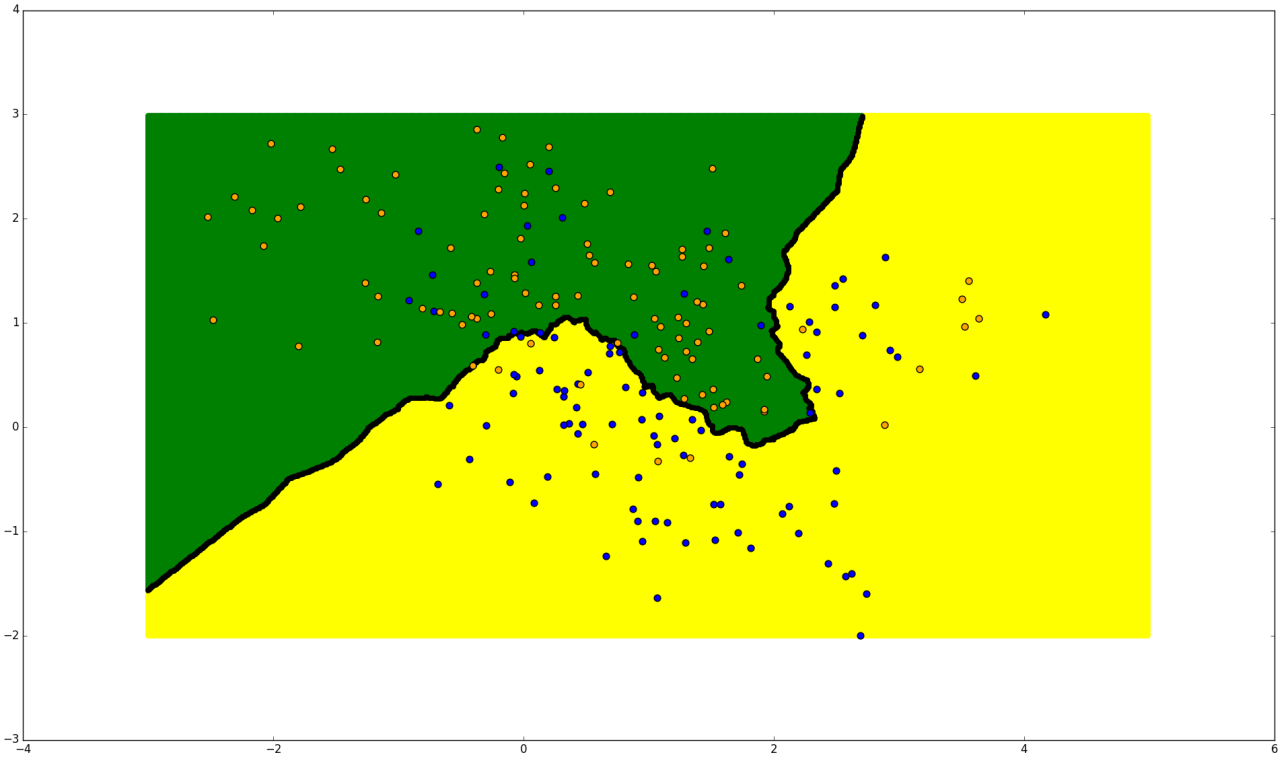

Figure 5: Classification using K-NN, $K = 15$ – The orange and blue circles represents the input points (Blue = Class 0, Orange = Class 1), The green and yellow represents the regions (with same respective class labels) and the black boundary in between the regions represent the classification boundary obtained using this method

(c) Confusion Matrix

Figure 6 shows the confusion matrix values for least squares classification. Similarly, Figure 7 and Figure8 shows the confusion matrix values for K-Nearest Neighbors classification using $K = 1$ and $K = 15$ respectively.

The basic difference between the least squares method and the K-nearest neighbor is that the former assumes a linear model and hence the classification regions and the boundaries may be more erroneous for new test data. Whereas the latter does not assume a linear model and hence is free of errors due to linearity assumption. Within the K-NN themselves, we can notice a lot of difference in performance based on the K value. For $K = 1$ we get 100% classification rate, but this will definitely cause a huge error while testing, because the regions are based on the classes in the training data itself. In contrast, for $K = 15$, we get a better boundary, and hence the error during testing is reduced. However, if we increase K to a large value, the error will still increase because the space for regions will become lesser and lesser and at some point of time the regions for two classes will overlap! Hence, it is important to choose an optimal value for K. This could be an advantage of the linear model, since it is not based on any external parameters such as K.

	Class 0	Class 1
Class 0	70	24
Class 1	30	76

Figure 6: Confusion matrix using least-squares method – Table shows the number of points classified as their true class and that are mistakenly classified (False classification).

	Class 0	Class 1
Class 0	100	0
Class 1	0	100

Figure 7: Confusion matrix using K-NN, $K = 1$ – Table shows the number of points classified as their true class and that are mistakenly classified (False classification).

	Class 0	Class 1
Class 0	82	13
Class 1	18	87

Figure 8: Confusion matrix using K-NN, $K = 15$ – Table shows the number of points classified as their true class and that are mistakenly classified (False classification).

Bibliography

[1] Jerome Friedman, Trevor Hastie, and Robert Tibshirani. *The elements of statistical learning*, volume 1. Springer series in statistics Springer, Berlin, 2001.

[2] P Ganesan, E M Cherry, A M Pertsov, and B Ghoraani. Characterization of electrograms from multipolar diagnostic catheters during atrial fibrillation. *BioMed Research International*, 2015, 2015.

[3] Jose Jalife. Rotors and spiral waves in atrial fibrillation. *Journal of cardiovascular electrophysiology*, 14(7):776–780, 2003.

List of Figures