Boğaziçi University

Cmpe462

Machine Learning

HW2 REPORT

Author:
Enes Özipek

Professor:
Emre Uğur

April 2, 2017

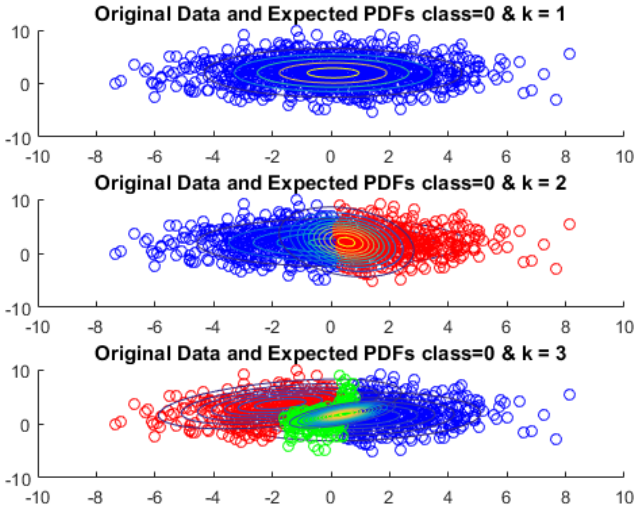
1 Description

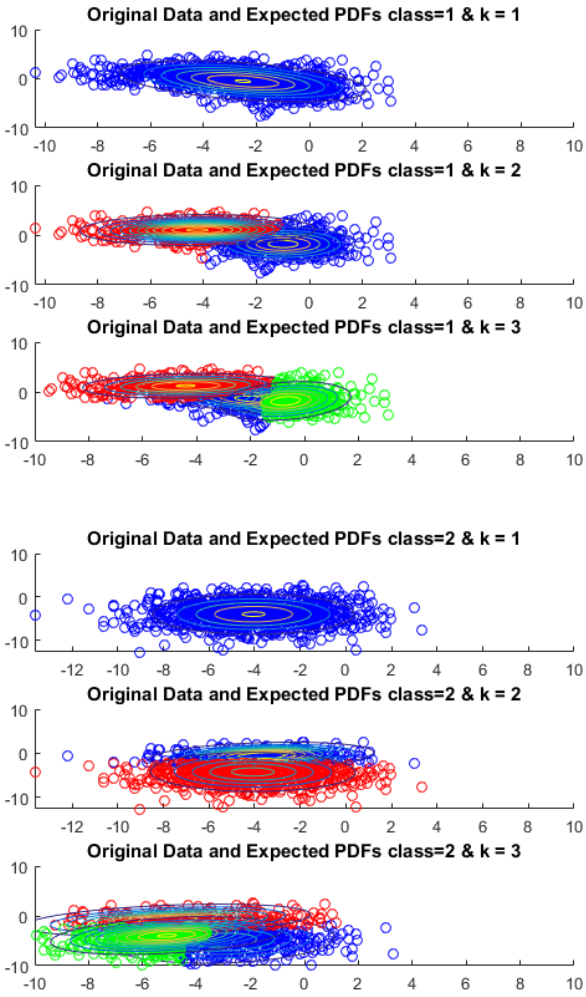
We are given a set of multivariate data classified by 3 classes, 0,1 and 2. We are expected to guess class of a new-coming data by using semi-parametric and non-parametric method, respectively.

2 Methods

2.1 Mixtures of Gaussians

Algorithm starts with selecting k data points to serve as a initial means which is calculated by using kmeans method. Cluster probabilities and covariance of the clusters are initialized. Then we use a matrix to hold the probability that each data point belongs to each cluster. We declare a convergence condition to check. We calculate the probability for each data point for each cluster. To accomplish that we use a matrix to hold the pdf values for each data point for each cluster. We calculate contribution of each each data points and then we calculate normalized posterior probability for each data point, in other words, we show how probable a data point can be in a spesific cluster.





2.1.1 Number of Gaussians

To calculate the number of gaussians for each class(for the best model), I consider the correct guess number made for each cluster. By considering the ratio, we can conclude that correct guesses made for each class generally falls into 3 cluster model.

Best k for class 0: 3, ratio(%): 80.16
Best k for class 1: 3, ratio(%): 79.56
Best k for class 2: 3, ratio(%): 94.36

2.1.2 Prediction Error

To get the following result, I consider the diagonal entries of confusion matrix since it denotes true-true values.

Prediction Error(%) for each class respectively:
class 0: 25.4000
class 1: 26.4000
class 2: 25.0000

2.1.3 The confusion matrix

The confusion matrix entries denotes values true-true and true-false. Yet, to be able to show all classes in one matrix I utilized the following logic:

- confusion(1,1) $\rightarrow$ class0(guess) & class0(real)
- confusion(1,2) $\rightarrow$ class0(guess) & class1(real)
- confusion(1,3) $\rightarrow$ class0(guess) & class2(real)

$\vdots$

- confusion(3,1) $\rightarrow$ class2(guess) & class0(real)
- confusion(3,2) $\rightarrow$ class2(guess) & class1(real)
- confusion(3,3) $\rightarrow$ class2(guess) & class2(real)

$$\text{confusion} = \begin{bmatrix} 373 & 115 & 12 \\ 67 & 368 & 65 \\ 10 & 115 & 375 \end{bmatrix}$$

2.2 K-NN

In KNN implementation, we consider first batch of the points as training set. Then other batch of the points are for test set. For each point in test set, we calculate eucledian distance to each point in the training set. Then we sort the distances in ascending order. We consider the classes of first K points. Then we take the mode of those points with respect to their classes. Most frequent class will be the candidate class for our test point.

2.2.1 Best k

Best K is selected by considering the correct guess number made i that K.
Best k: 40

2.2.2 Prediction Error

Prediction Error(%) for each k:

$$\begin{gathered} \text{K:1} \rightarrow 32.6333 \\ \text{K:10} \rightarrow 25.2333 \\ \text{K:40} \rightarrow 23.8333 \end{gathered}$$

2.2.3 Confusion Matrix

Logic is as the following:

- confusion(1,1) $\rightarrow$ k1(guess) & k1(real)
- confusion(1,2) $\rightarrow$ k1(guess) & Not k1

$$\vdots$$

- confusion(3,1) $\rightarrow$ k40(guess) & k40(real)
- confusion(3,2) $\rightarrow$ k40(guess) & Not k40

$$\text{confusion} = \begin{bmatrix} 2021 & 979 \\ 2243 & 757 \\ 2285 & 715 \end{bmatrix}$$

3 Util Functions

3.1 Gaussian Function

I also wrote my gaussian probability calculation method. It's basically calculates pdf of the given matrix with given mu and sigma.