CS 584

MACHINE LEARNING

ASSIGNMENT 2

by

Saurabh Katkar

A20320336

Code created using Rstudio version 0.98.1062

Q.1 Clustering

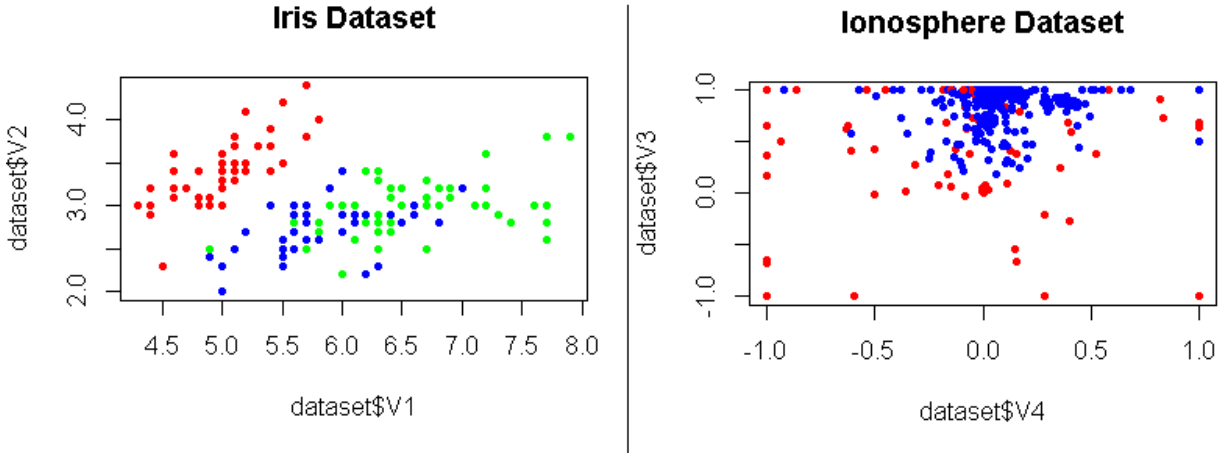
1.

Datasets used:

- Iris Dataset: (150 instances, 5 attributes, 3 classes)
- Ionosphere Dataset (351 instances, 35 attributes, 2 classes)

Labels of the datasets (ie, the final column) were not considered in the code.

Dataset Plots:



2.

K means algorithm implemented in our code:

1. Decide on the number of clusters.
2. Initialize the center of the clusters

   μ_i= some value, i=1,...,k
3. Attribute the closest cluster to each data point

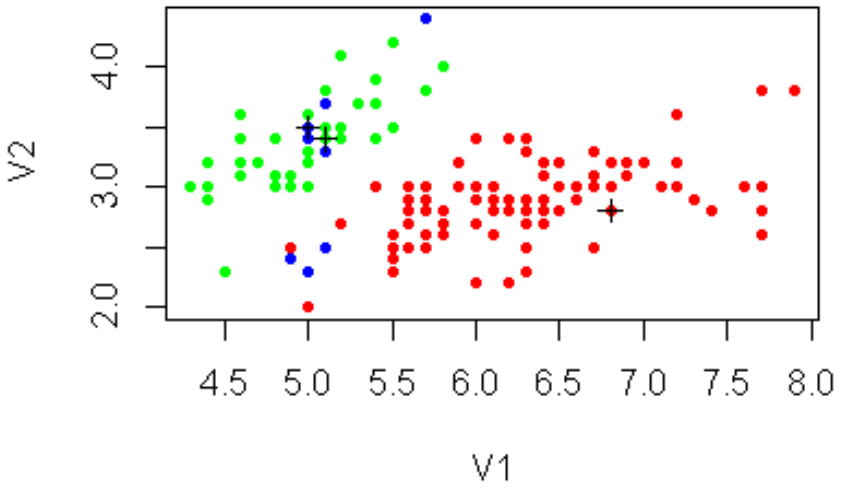
   $\mathbf{c}i=\{j:d(\mathbf{x}j,\mu i)\leq d(\mathbf{x}j,\mu l),l\neq i,j=1,...,n\}$
4. Set the position of each cluster to the mean of all data points belonging to that cluster

   $\mu i=1|ci|\sum j\in ci\mathbf{x}j,\forall i$
5. Repeat steps 2-3 until convergence

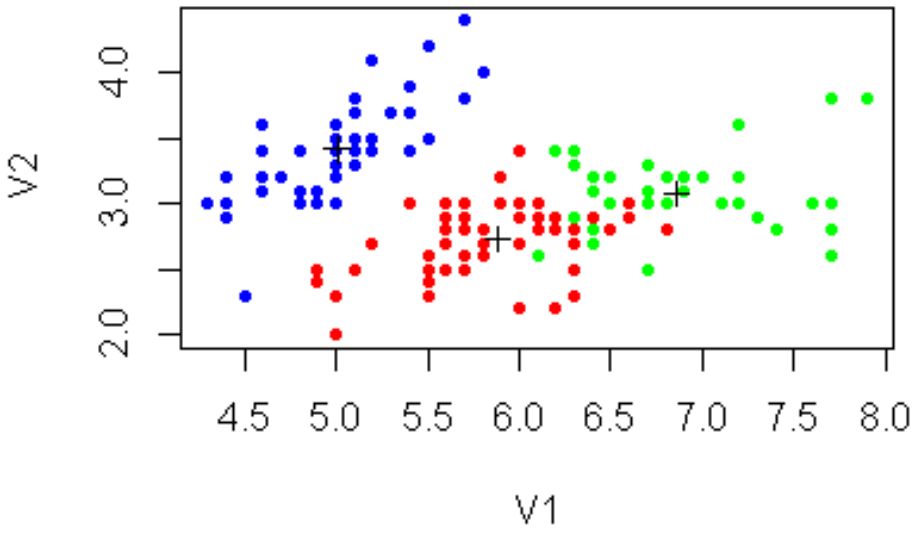
Clusters formed using K-means Algorithm:

For Iris Dataset

After 1 iteration:

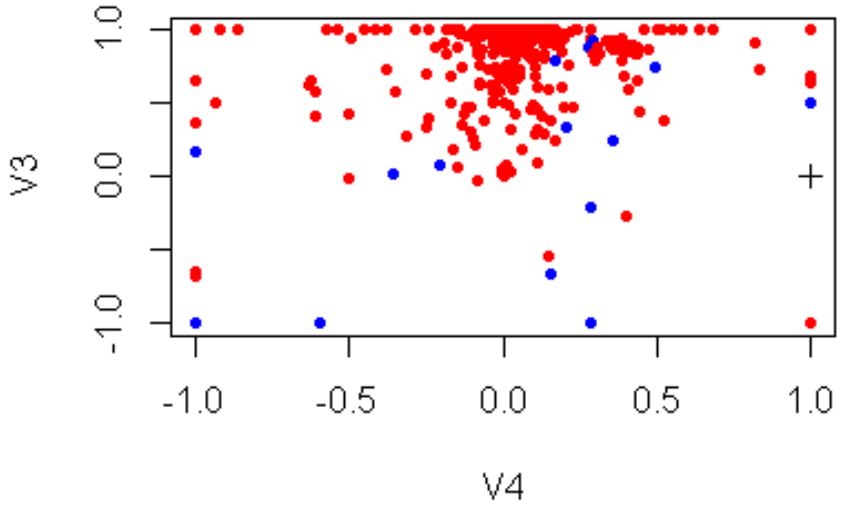


After 100 iterations:

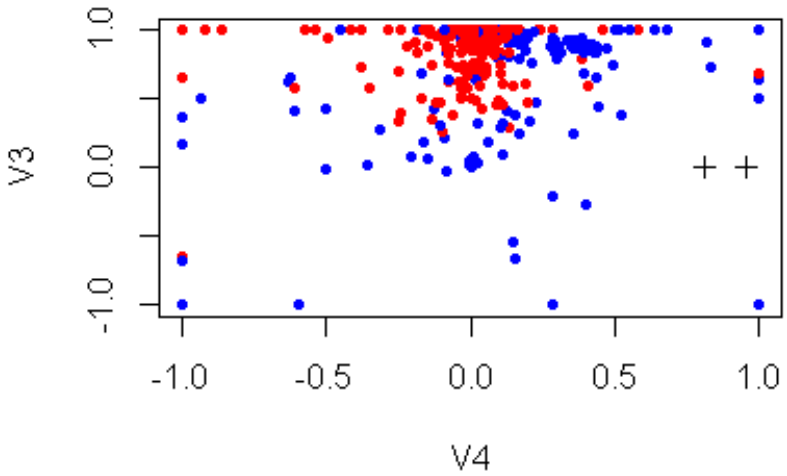


For Ionosphere Dataset:

After 1 iteration:



After 100 iterations:

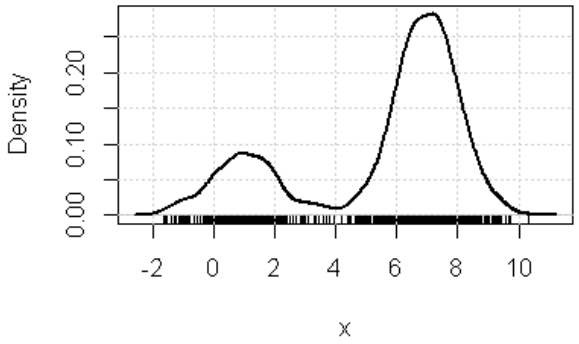


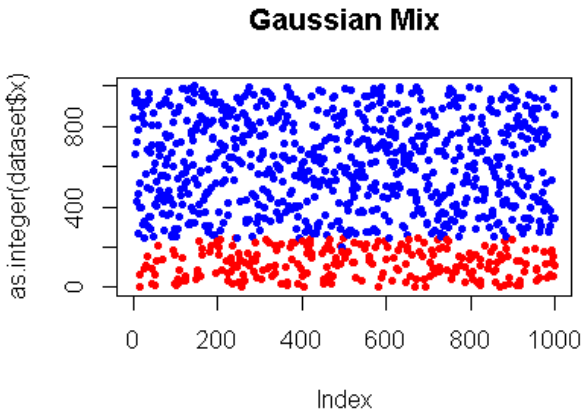
3.

Implementing EM algorithm using a Gaussian mixture

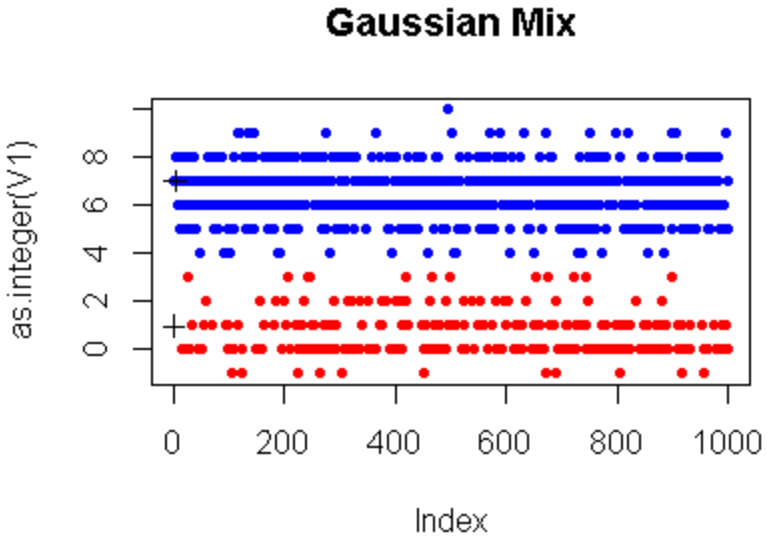
Self – generated sGaussian mixture dataset

Dataset Plots obtained:





Plot obtained after applying EM algorithm using Gaussian mixture:



4.

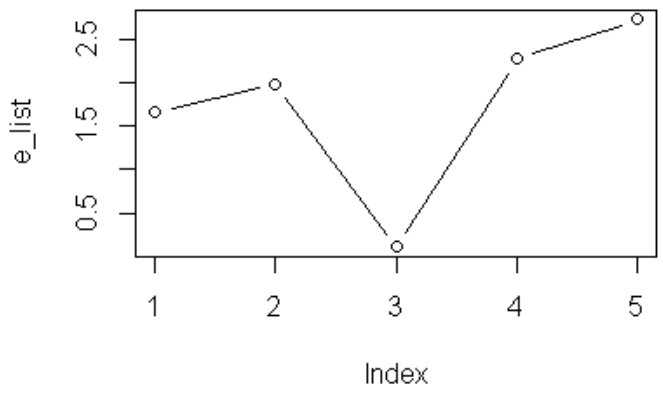
Method for automatically determining the number of clusters:

- Perform K means algorithm on different number of clusters.
- After executing K means, observe the mean square error obtained by comparing the derived labels with the known labels
- Choose cluster that gives the lowest mean square error while using this method

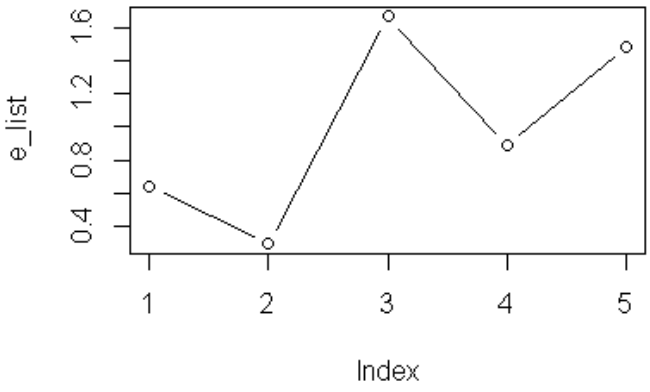
Plots derived by using this method:

(Index denotes the number of clusters)

For Iris data:



For Ionosphere Data:



5.

Error Rates

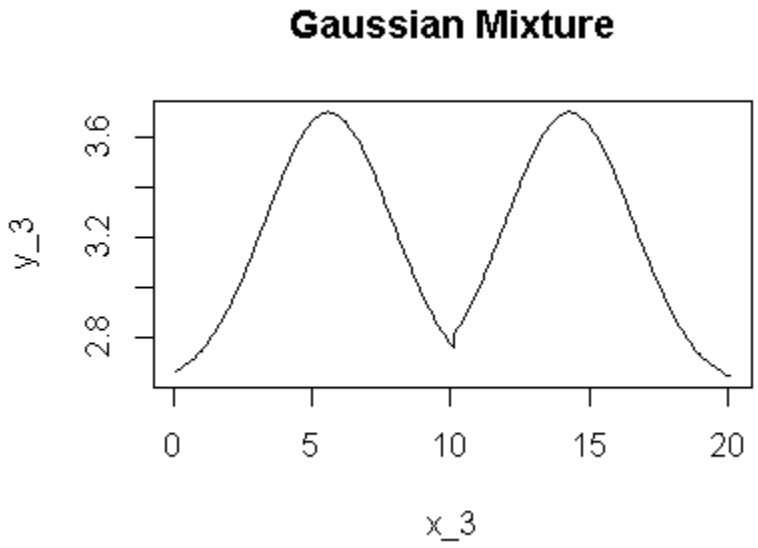
Mean Square Error	
Dataset	**Kmeans**
Iris	0.107
Ionosphere	0.291

Mean Square Error	
Dataset	**EM**
Gaussian Mix	0.002

Q.2 Factor Analysis

1.

Data Plots of 2D Gaussians clusters



2.

Performing PCA dimensionality reduction

We produce a compact low-dimensional encoding of a given high-dimensional data set. PCA dimensionality reduction can be performed on datasets which have a large number of features.

In our code, using dimensionality increase, we increase the dimensions of our Gaussian dataset. Then Principal components analysis (PCA) is used that provides a sequence of best linear approximations to a given high-dimensional observation.

3.

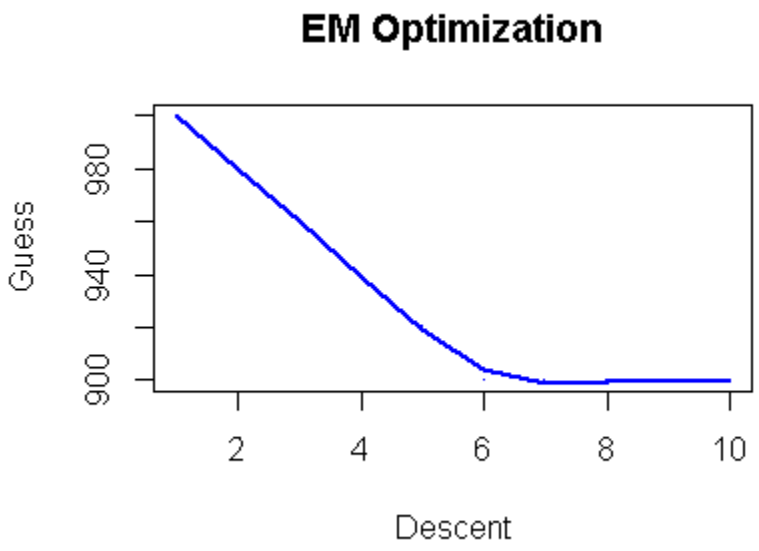
Applying EM Factor analysis and plotting the factors

EM Factor analysis used for explaining data, in particular, correlations between variables in multivariate observations and for dimensionality reduction.

In EM factor analysis, we initially guess the parameters which are the Latent variables, the mean and the standard deviation.

Using the guesses we determine the Probability of a cluster to which the point might belong to and the probability that a point belongs to a particular cluster.

Next maximize the parameters and once the parameters are optimized find the probabilities and find the maximum of the computed probabilities.



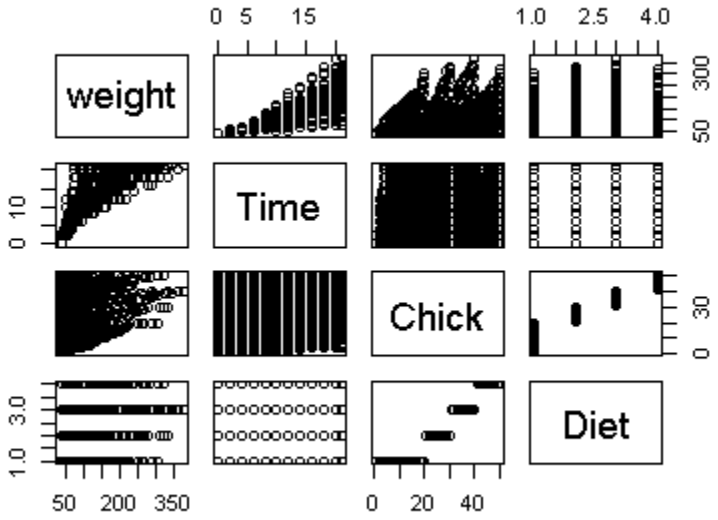
4.

Datasets selected: R built-in

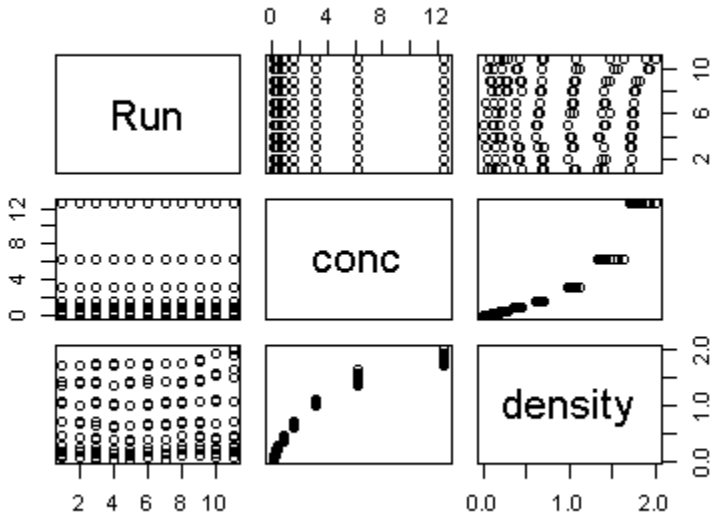
Chickweight (578*4, 4 classes)

DNase (176*3, 11 classes)

Chicweight plot:



DNase plot:



6.

Automatically determining the number of factors:

We choose principal component to be the smallest value so that;

$$\frac{1/m\sum_i |x - x_{approx}|^2}{1/m\sum_i |x|^2} < 0.01$$

So that in this case 99% variance is retained

Algorithm:

1. Try PCA with k=1,2,3…
2. Compute Parameters: U, z, x
3. Check if $\frac{1/m\sum_i |x - x_{approx}|^2}{1/m\sum_i |x|^2} < 0.01$ ie. 99% variance is retained
4. Choose number of factors that retain that variance