ST: Statistics Project Report
Supervised Classification

Talha Mahmood Chaudhry
U00726473

November 20, 2020

Introduction

In this report the analysis in the paper titled mclust 5: Clustering, Classification and Density Estimation Using Gaussian Finite Mixture Models for **Supervised Classification** is explained and reproduced. This has been done using the **R** software, in particular the *mclust* package. It is a popular **R** package for model-based clustering, classification, and density estimation. This package allows modelling of data as a Gaussian finite mixture with different covariance structures and different numbers of mixture components. The dataset to be used to show this analysis is the UCI Wisconsin breast cancer diagnostic data which is easily available online. All figures and the entire code can be seen in the Appendix.

Theoretical Framework

In Supervised Classification (or discriminant analysis) the objective is to build a classifier that is able to assign an unclassified observation to one of the known K classes.

The supervised classifier is built using a training data set $\{(\mathbf{x_1}, \mathbf{y_1}), \ldots, (\mathbf{x_n}, \mathbf{y_n})\}$, where $\mathbf{x_i}$ are the features and the true classes $\mathbf{y_i} \in \{1, \ldots, K\}$ are known. It is assumed that the density of each class has a Gaussian mixture distribution:

$$f_k(\mathbf{x}) = \sum_{g=1}^{G_k} \pi_{gk} \phi(\mathbf{x}; \boldsymbol{\mu}_{gk}, \boldsymbol{\Sigma}_{gk})$$

- $\boldsymbol{\pi}_{g\mathbf{k}}$ are the mixing probabilities of class k, $\pi_{gk} > 0$ and $\sum_{g=1}^{G_k} \pi_{gk} = 1$.
- $\boldsymbol{\mu}_{gk}$ are the means for component g in class k.
- $\boldsymbol{\Sigma}_{gk}$ is the covariance matrix of component g within class k.

Now the Discriminant Analysis can be subdivided into two broad approaches:

1. Mixture Discriminant Analysis[1] (MDA) where where the only constrained assumption is that the covariance matrix is equal for all components g in class k, i.e., $\mathbf{\Sigma}_{gk} = \mathbf{\Sigma}$. It is also assumed all g are known for each k.
2. Eigenvalue Decomposition Discriminant Analysis[2] (EDDA) where it is assumed that the density of each class can be described by a single Gaussian component, i.e, $G_k = 1$ for all k, and the covariance matrix can be factorized.

Eigenvalue Decomposition of the Covariance Matrices, $\mathbf{\Sigma_k}$

$$\mathbf{\Sigma_k} = \lambda_\mathbf{k}\mathbf{D_k}\mathbf{A_k}\mathbf{D'_k},$$

,

- $\mathbf{D_k}$ is a matrix of eigenvectors of $\mathbf{\Sigma_k}$.
- $\mathbf{A_k} = Diag\{A_{1,k}, \ldots, A_{d,k}\}$ is a diagonal matrix where the elements are proportional to the eigenvalues of Σ_k in descending order. (scaled)
- $\lambda_\mathbf{k}$ is the constant of proportionality.

In the multivariate case this results in 14 different possible models based on the decomposition as shown in Figure **1**. Now, for example, consider model **EEE**, this is where the constants of proportionality, the eigenvectors, and the eigenvalues are equal , i.e. $\mathbf{\Sigma_k} = \lambda\mathbf{DAD'}$, EDDA is akin to Linear Discriminant Analysis (LDA). Recall that an important assumption of the LDA is *homoscedasticity*, that is the homogeneity of variance/covariance, where variance among the components and covariance across the classes are equal. Similarly, in model **EII** the constant of proportionality is equal, and the eigenvector matrix **D** and the diagonal scaled eigen value matrix **A** are not only equal but are in fact the Identity matrix. Thus, variances are equal and 1, and the covariances are equal and zero. Geometrically this means that the shape of the clusters will be spherical of equal volume (in $\mathbb{R}^d$) for **EII** and ellipsoidal of equal volume for **EEE** (The variance/covariances are equal but not 1 thus stretching or contracting the shape in space).
In contrast, model **VVV** is akin to Quadratic Discriminant Analysis, that is $\mathbf{\Sigma_k} = \lambda_\mathbf{k}\mathbf{D_k}\mathbf{A_k}\mathbf{D'_k}$. This means that the ellipsoidal clusters will have differing volumes, shapes, and alignment. This is easy to see since for QDA there are no constraints on the variances and covariances.

Therefore, the Taxonomy of the models is done the following way. For the first letter, if $\lambda_\mathbf{k}$ are equal then volume is **E**qual and **V**ariable otherwise. For

[1] Hastie and Tibshirani (1996)
[2] Bensmail and Celeux (1996)

the second letter, if the shape matrix $\mathbf{A_k}$ is equal across clusters then letter **E** (ellipsoidal of same shape), and **V** otherwise (ellipsoidal of different shape). If it is constrained to be **I**, which signifies the identity matrix, then the shape is spherical. Finally, for the third letter $\mathbf{D_k}$, the alignment matrix, plays the pivotal role. If it is constrained to **I**, then that signifies the clusters are *axis*-aligned. If they are **E**qual then they have the same orientation but not axis-alligned. In case when **V**ariable, then they are not axis-alligned nor have the same orientation. (See Fig **2** for a snapshot)

Data

The Wisconsin Breast Cancer Diagnostic data has record of 569 Patients with 30 features of the cell nuclei obtained from a digitized image of a fine needle aspirate (FNA) of a breast mass[3]. There are two classes: Benign (357) vs Malignant (212). $2/3^{rd}$of the data will be used for training and the rest will be used for testing. The following Features[4] will be in consideration:

1. Extreme area
2. Extreme smoothness
3. Mean texture

(Note: See Figure **3** for a visualization)

Building Classifiers

As stated earlier the data is split into training ($2/3^{rd}$) and testing set ($1/3^{rd}$). The following is a tabulation of the Classes as they categorize in these two sets:

B	M	B	M
238	142	119	70

The function MclustDA() from the *mclust* package is used to build the classifier using EDDA by specifying the optional argument modelType = "EDDA". This forces single component mixtures across classes. The following is the output:

[3]Mangasarian et al., 1995
[4]Following Fraley and Raftery (2002)

```
> mod1 <- MclustDA(X.train, Class.train, modelType = 'EDDA')
fitting ...
  |============================================================
> summary(mod1, newdata = X.test, newclass = Class.test)
-----------------------------------------------
Gaussian finite mixture model for classification
-----------------------------------------------

EDDA model summary:

 log-likelihood   n df       BIC
      -2964.242 380 12 -5999.767

Classes   n     % Model G
      B 238 62.63   VVI 1
      M 142 37.37   VVI 1

Training confusion matrix:
     Predicted
Class   B   M
    B 232   6
    M  11 131
Classification error = 0.0447
Brier score          = 0.0349

Test confusion matrix:
     Predicted
Class   B   M
    B 119   0
    M   9  61
Classification error = 0.0476
Brier score          = 0.0328
```

The EDDA mixture model selected by BIC is the **VVI** model, so each group is described by a single Gaussian component with differing volume and shape, but orientation is axis-aligned. The EDDA imposed a single mixture component for each class. See Figure **4)** for a visualization of this with pairwise scatterplots between the features, showing both the known classes and the estimated mixture components. Notice there are only one cluster each which are axis-aligned however the shapes and areas are different.A cross-validation error can also be computed using the cvMclust() function:

```
> cv <- cvMclustDA(mod1)
cross-validating ...
  |============================
> unlist(cv[c('error', 'se')])
     error         se
0.04736842 0.01017127
```

However, now a more general approach[5] can be made using the MclustDA() function where a finite mixture of Gaussian components is allowed in each class. The following is the output:

```
> mod2 <- MclustDA(X.train, Class.train)
Class B: fitting ...
  |============================================================
Class M: fitting ...
  |============================================================
> summary(mod2, newdata = X.test, newclass = Class.test)
------------------------------------------------
Gaussian finite mixture model for classification
------------------------------------------------

MclustDA model summary:

 log-likelihood   n df       BIC
      -2928.494 380 24 -5999.552

Classes   n     % Model G
      B 238 62.63   VEI 2
      M 142 37.37   VVI 2

Training confusion matrix:
     Predicted
Class   B   M
    B 232   6
    M   6 136
Classification error = 0.0316
Brier score          = 0.0276

Test confusion matrix:
     Predicted
Class   B   M
    B 119   0
    M   5  65
Classification error = 0.0265
Brier score          = 0.0269
```

The classification errors are lower than the first model. This time around the model picked $G = 2$ components for each class. For the "Benign" class the decomposition picked **VEI**, that is the area are different but the shape is the same and the two clusters are axis-aligned. For the "Malignant" class both the area

[5] Fraley and Raftery (2002)

and shape were variable but axis-aligned, which is **VVI**. See Figure **5(a)(b)(c)** for the visualization of pairwise scatterplots. Cross-validation error were also be computed using the cvMclust() function:

```
> cv <- cvMclustDA(mod2)
cross-validating ...
  |============================
> unlist(cv[c('error', 'se')])
     error         se
0.03947368 0.01059306
```

The cv-error is lower than the previous model suggesting a better model.
Another interesting graph can be obtained by projecting the data on a dimension reduced subspace[6] using the previous model with the following output (see Figure **5(d)**:

```
> drmod2 <- MclustDR(mod2)
> summary(drmod2)
-----------------------------------------------------------------
Dimension reduction for model-based clustering and classification
-----------------------------------------------------------------

Mixture model type: MclustDA

Classes   n Model G
      B 238   VEI 2
      M 142   VVI 2

Estimated basis vectors:
                          Dir1        Dir2
texture.mean       -9.9808e-01  9.9999e-01
area.extreme       -6.2015e-02 -3.4189e-03
smoothness.extreme -6.1725e-05  6.1279e-05

               Dir1      Dir2
Eigenvalues  0.65213   0.16662
Cum. %      79.64932 100.00000
```

Two directional basis vectors encompass most of the data. The two groups are largely separated along the first direction, with the group of malignant cases showing a higher variability.

Finally, MDA can be performed, this is similar to LDA where the model is **EEE**. On the basis of the second model the components in each class are constrained to $G = 2$ each, so 2 clusters for each class. Following is the output:

[6] Scrucca (2014)

```
> mod3 <- MclustDA(X.train, Class.train, G = 2, modelNames = 'EEE')
Class B: fitting ...
  |============================================================
Class M: fitting ...
  |============================================================
> summary(mod3, newdata = X.test, newclass = Class.test)
-----------------------------------------------
Gaussian finite mixture model for classification
-----------------------------------------------

MclustDA model summary:

 log-likelihood   n df      BIC
      -2953.803 380 26 -6062.05

Classes   n     % Model G
      B 238 62.63   EEE 2
      M 142 37.37   EEE 2

Training confusion matrix:
     Predicted
Class   B   M
    B 232   6
    M   8 134
Classification error = 0.0368
Brier score          = 0.0255

Test confusion matrix:
     Predicted
Class   B   M
    B 119   0
    M   6  64
Classification error = 0.0317
Brier score          = 0.0278
```

The classification error is really close to the second model, please see Figure 6 for pairwise graph. All the clusters for both the classes have equal shape, area, and orientation.

Concluding Remark

The *mclust* package is really useful for modelling and clustering with data and its analysis. However, it only works with Gausssian assumptions for the underlying distribution. But, overall, it has many applications and deserves its popularity in recent years.

Refrences

1. Hastie T, Tibshirani R. Discriminant analysis by Gaussian mixtures. Journal of the Royal Statistical Society: Series B (Statistical Methodology). 1996; 58(1):155–176.

2. Bensmail H, Celeux G. Regularized Gaussian discriminant analysis through eigenvalue decomposition. Journal of the American Statistical Association. 1996; 91:1743–1748.

3. Mangasarian OL, Street WN, Wolberg WH. Breast cancer diagnosis and prognosis via linear programming. Operations Research. 1995; 43(4):570–577.

4. Fraley C, Raftery AE. Model-based clustering, discriminant analysis, and density estimation. Journal of the American Statistical Association. 2002; 97(458):611–631.

5. Scrucca L. Graphical tools for model-based mixture discriminant analysis. Advances in Data Analysis and Classification. 2014; 8(2):147–165. http://dx.doi.org/10.1007/s11634-013-0147-1.

Appendix

Figure 1: Table of 14 possible models

EII	λI	Spherical	Equal	Equal	NA
VII	$\lambda_g I$	Spherical	Variable	Equal	NA
EEI	λA	Diagonal	Equal	Equal	Axis-aligned
VEI	$\lambda_g A$	Diagonal	Variable	Equal	Axis-aligned
EVI	λA_g	Diagonal	Equal	Variable	Axis-aligned
VVI	$\lambda_g A_g$	Diagonal	Variable	Variable	Axis-aligned
EEE	Σ	Ellipsoidal	Equal	Equal	Equal
VEE	$\lambda_g D A D^T$	Ellipsoidal	Variable	Equal	Equal
EVE	$\lambda D A_g D^T$	Ellipsoidal	Equal	Variable	Equal
EEV	$\lambda D_g A D_g^T$	Ellipsoidal	Equal	Equal	Variable
VVE	$\lambda_g D A_g D^T$	Ellipsoidal	Variable	Variable	Equal
EVV	$\lambda D_g A_g D_g^T$	Ellipsoidal	Equal	Variable	Variable
VEV	$\lambda_g D_g A D_g^T$	Ellipsoidal	Variable	Equal	Variable
VVV	Σ_g	Ellipsoidal	Variable	Variable	Variable

Figure 2: An example

Figure 3: Boxplot of the Features

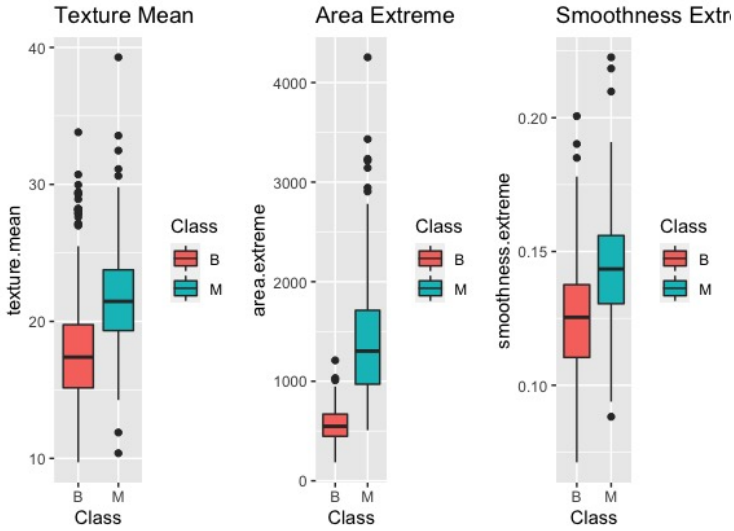


Figure 4: Pairwise Scatterplot with Model 1

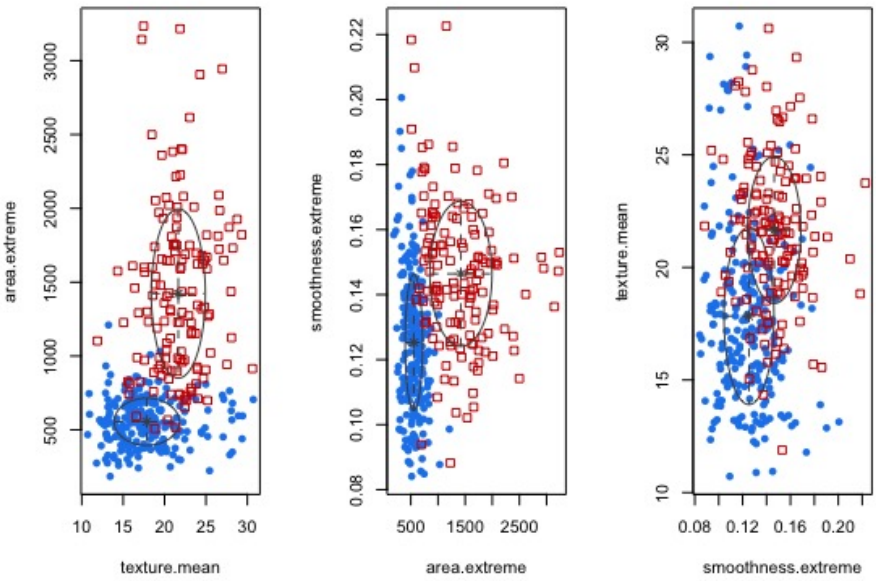


Figure 5: Pairwise Scatterplot with Model 2, Including Dimension Reduction

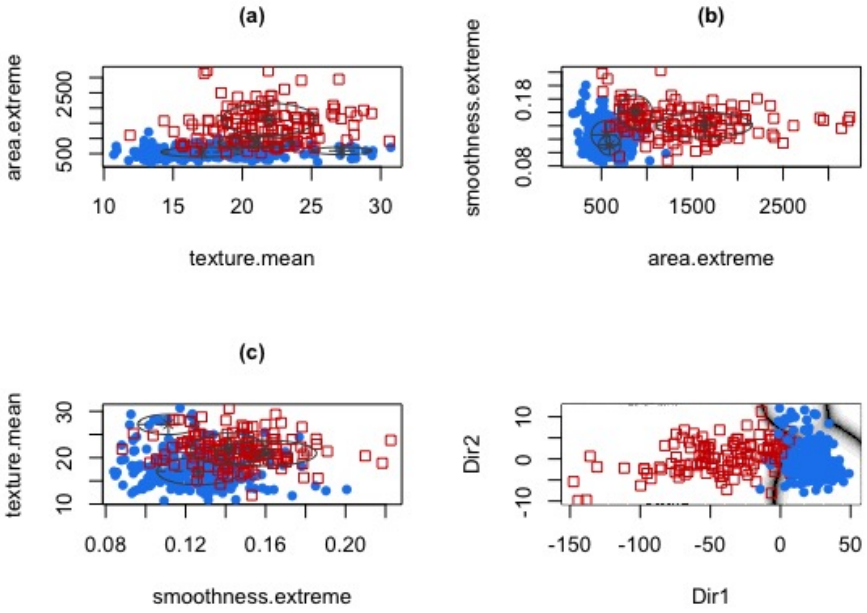
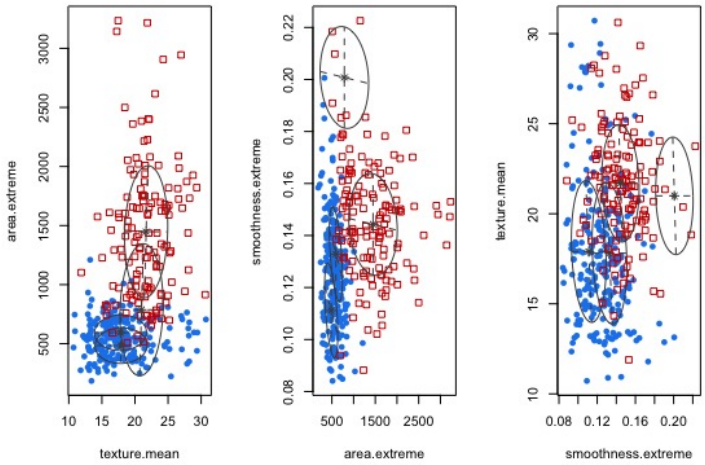


Figure 6: Pairwise Scatterplot with Model 3



R Code

```
library(mclust);library(caret); library(ggplot2); library(gridExtra)
filepath <- "http://archive.ics.uci.edu/ml/machine-learning-databases/
breast-cancer-wisconsin/wdbc.data"


my_data <- read.csv(filepath, header = FALSE)


X <- my_data[,c(4, 26, 27)]


colnames(X) <- c('texture.mean', 'area.extreme', 'smoothness.extreme')


Class <- my_data[,2]

par(mfrow = c(1,3))

g1 <- ggplot(X, aes(x=Class, y=texture.mean, fill=Class)) +
  geom_boxplot() + ggtitle("Texture Mean")
g2 <- ggplot(X, aes(x=Class, y=area.extreme, fill=Class)) +
  geom_boxplot() + ggtitle("Area Extreme")
g3 <- ggplot(X, aes(x=Class, y=smoothness.extreme, fill=Class)) +
  geom_boxplot() + ggtitle("Smoothness Extreme")
grid.arrange(g1, g2, g3, nrow = 1)

set.seed(123)

intrain <- createDataPartition(Class, p = 2/3, list = FALSE)
X.train <- X[intrain, ]
Class.train <- Class[intrain]

pander(table(Class.train))

X.test <- X[-intrain, ]
Class.test <- Class[-intrain]

pander(table(Class.test))

mod1 <- MclustDA(X.train, Class.train, modelType = 'EDDA')

summary(mod1, newdata = X.test, newclass = Class.test)
```

```
cv <- cvMclustDA(mod1)
unlist(cv[c('error', 'se')])

par(mfrow = c(2,2))
plot(mod1, what = 'scatterplot', dimens = c(1,2))
plot(mod1, what = 'scatterplot', dimens = c(2,3))
plot(mod1, what = 'scatterplot', dimens = c(3,1))

mod2 <- MclustDA(X.train, Class.train)
summary(mod2, newdata = X.test, newclass = Class.test)

cv <- cvMclustDA(mod2)
unlist(cv[c('error', 'se')])

par(mfrow = c(1,4))
plot(mod2, what = 'scatterplot', dimens = c(1,2), main = '(a)')
plot(mod2, what = 'scatterplot', dimens = c(2,3), main = '(b)')
plot(mod2, what = 'scatterplot', dimens = c(3,1), main = '(c)')

drmod2 <- MclustDR(mod2)
summary(drmod2)
par(mfrow = c(1,1))
plot(drmod2, what = 'boundaries', ngrid = 200, main = '(d)')

mod3 <- MclustDA(X.train, Class.train, G = 2, modelNames = 'EEE')
summary(mod3, newdata = X.test, newclass = Class.test)
par(mfrow = c(1,3))
plot(mod3, what = 'scatterplot', dimens = c(1,2))
plot(mod3, what = 'scatterplot', dimens = c(2,3))
plot(mod3, what = 'scatterplot', dimens = c(3,1))
```