Linear Discriminant Analysis (LDA)

By - Aaditya Gupta BT15CSE001

LDA is a supervised technique.

Linear Discriminant Analysis can be seen from two different angles:

1. The first classify a given sample of predictors x to the class C_l with highest posterior probability $\pi(y = C_l|x)$. It minimises the total probability of misclassification. To compute $\pi(y = C_l|x)$ it uses Bayes' rule and assume that $\pi(x|y = C_l)$ follows a Gaussian distribution with class-specific mean μ_l and common covariance matrix Σ.
2. The second tries to find a linear combination of the predictors that gives maximum separation between the centers of the data while at the same time minimising the variation within each group of data.

The second approach is usually preferred in practice due to its dimension-reduction property and is implemented in many R packages, as in the `lda` function of the `MASS` package for example.

Practical

Data set Used - Iris

The call to lda contains the following arguments:

1. Formula
2. Data
3. Prior

```
r <- lda(formula = Species ~ .,data = iris,prior =c(1,1,1)/3)
```

The `.` in the `formula` argument means that we use all the remaining variables in `data` as covariates

The `prior` argument sets the prior probabilities of class membership. If unspecified, the class proportions for the training set are used. If present, the probabilities should be specified in the order of the factor levels.

```
> r$prior
    setosa versicolor  virginica 
 0.3333333  0.3333333  0.3333333 
> r$counts
    setosa versicolor  virginica 
        50         50         50 
> r$means
           Sepal.Length Sepal.Width Petal.Length Petal.Width
setosa            5.006       3.428        1.462       0.246
versicolor        5.936       2.770        4.260       1.326
virginica         6.588       2.974        5.552       2.026
> r$scaling
                    LD1         LD2
Sepal.Length  0.8293776  0.02410215
Sepal.Width   1.5344731  2.16452123
Petal.Length -2.2012117 -0.93192121
Petal.Width  -2.8104603  2.83918785
> r$svd
[1] 48.642644  4.579983
```

As we can see above, a call to `lda` returns

1. the `prior` probability of each class
2. the `counts` for each class in the `data`,
3. the class-specific `means` for each covariate,
4. the linear combination coefficients (`scaling`) for each linear discriminant
5. the singular values (`svd`) that gives the ratio of the between- and within-group standard deviations on the linear discriminant variables.

```
> prop = r$svd^2/sum(r$svd^2)
> prop
[1] 0.991212605 0.008787395
```

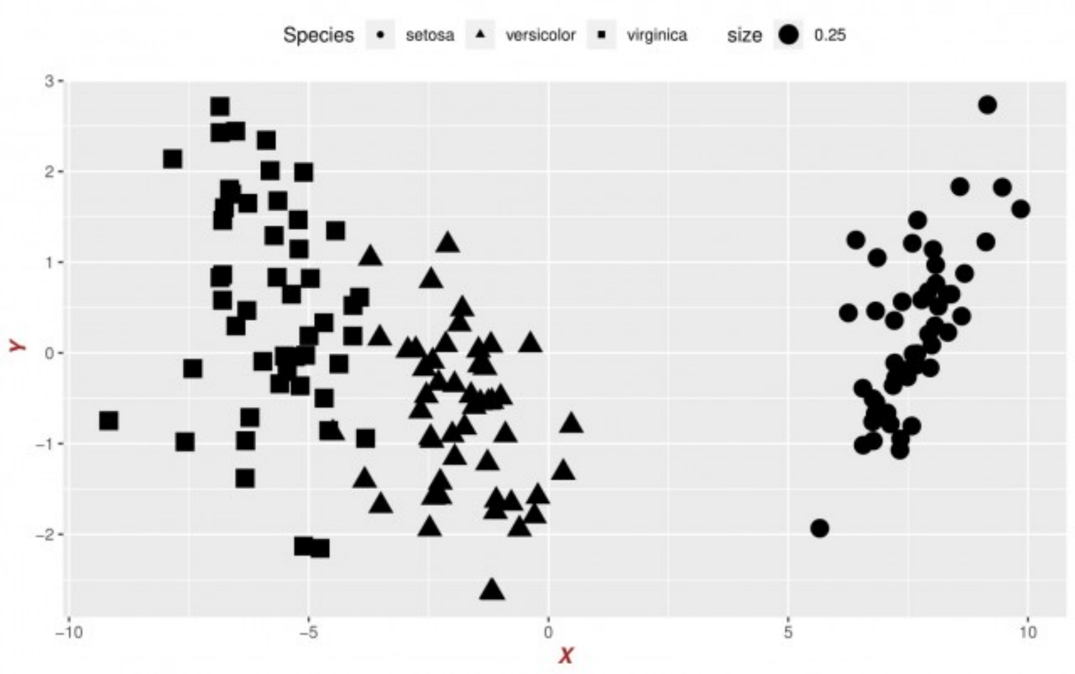
We can use the singular values to compute the amount of the between group variance that is explained by each linear discriminant.

If we call `lda` with `CV = TRUE` it uses a leave-one-out cross-validation and returns a named list with components.

Getting the cluster

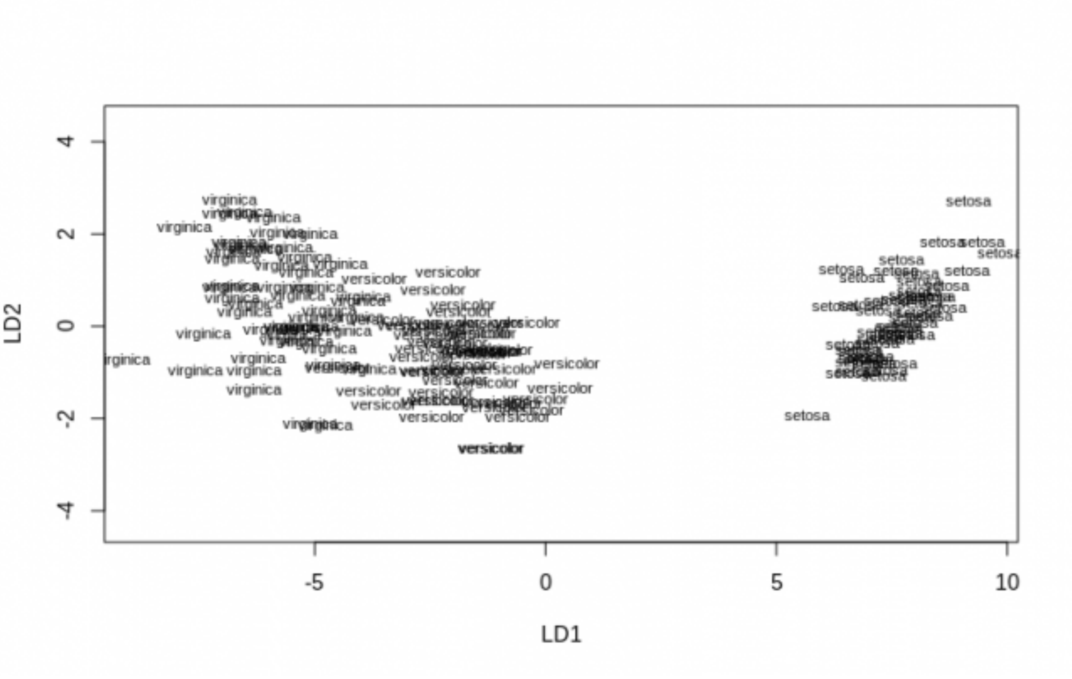
```
##This code can be directly reproduced in R Studio
require(MASS)
require(ggplot2)
data("iris")
my.data <- iris
head(my.data)
model <- lda(formula = Species ~ ., data = my.data)
data.lda.values <- predict(model)
plot.data <- data.frame(X=data.lda.values$x[,1], Y=data.lda.valu
head(plot.data)
p <- ggplot(data=plot.data, aes(x=X, y=Y)) +
  geom_point(aes(shape=Species,size=0.25,color=Species)) +
  theme(axis.title=element_text(face="bold.italic",
                                size="12", color="brown"), legen
p
```

The Output



We can also put the raw data on plot using

```
r <- lda(formula = Species ~ ., data = iris)
plot (r)
```


LD2
LD1
4
2
0
-2
-4
-5
0
5
10
virginica
versicolor
setosa