STHDA

Statistical tools for high-throughput data analysis

Search...

Home | Basics | Data | Visualize | Analyze | Products | Contribute

Support | About

Home / Articles / Machine Learning / Classification Methods Essentials / Discriminant Analysis Essentials in R

Articles - Classification Methods Essentials

Discriminant Analysis Essentials in R

kassambara | *11/03/2018* | 985 | *Comments (2)* | *Classification Methods Essentials*

Discriminant analysis is used to predict the probability of belonging to a given class (or category) based on one or multiple predictor variables. It works with continuous and/or categorical predictor variables.

Previously, we have described the logistic regression for two-class classification problems, that is when the outcome variable has two possible values (0/1, no/yes, negative/positive).

Compared to logistic regression, the discriminant analysis is more suitable for predicting the category of an observation in the situation where the outcome variable contains more than two classes. Additionally, it’s more stable than the logistic regression for multi-class classification problems.

Note that, both logistic regression and discriminant analysis can be used for binary classification tasks.

In this chapter, you’ll learn the most widely used discriminant analysis techniques and extensions. Additionally, we’ll provide R code to perform the different types of analysis.

The following discriminant analysis methods will be described:

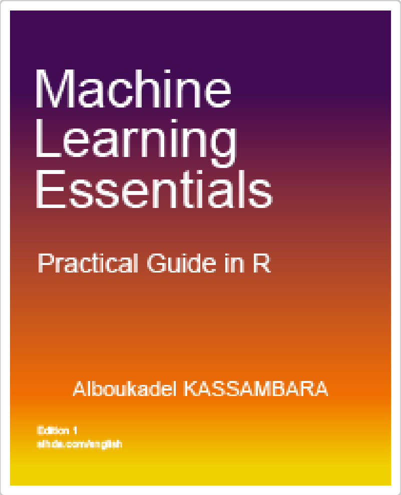
- **Linear discriminant analysis** (**LDA**): Uses linear combinations of predictors to predict the class of a given observation. Assumes that the predictor variables (p) are normally distributed and the classes have identical variances (for univariate analysis, p = 1) or identical covariance matrices (for multivariate analysis, p > 1).
- **Quadratic discriminant analysis** (**QDA**): More flexible than LDA. Here, there is no assumption that the covariance matrix of classes is the same.
- **Mixture discriminant analysis** (**MDA**): Each class is assumed to be a Gaussian mixture of subclasses.
- **Flexible Discriminant Analysis** (**FDA**): Non-linear combinations of predictors is used such as splines.

- **Regularized discriminant anlysis** (**RDA**): Regularization (or shrinkage) improves the estimate of the covariance matrices in situations where the number of predictors is larger than the number of samples in the training data. This leads to an improvement of the discriminant analysis.

Contents:

- Loading required R packages
- Preparing the data
- Linear discriminant analysis - LDA
- Quadratic discriminant analysis - QDA
- Mixture discriminant analysis - MDA
- Flexible discriminant analysis - FDA
- Regularized discriminant analysis
- Discussion
- References

The Book:



Machine Learning Essentials: Practical Guide in R

Loading required R packages

- tidyverse for easy data manipulation and visualization
- caret for easy machine learning workflow

```
library(tidyverse)
library(caret)
theme_set(theme_classic())
```

Preparing the data

We'll use the iris data set, introduced in Chapter @ref(classification-in-r), for predicting iris species based on the predictor variables Sepal.Length, Sepal.Width, Petal.Length, Petal.Width.

Discriminant analysis can be affected by the scale/unit in which predictor variables are measured. It's generally recommended to standardize/normalize continuous predictor before the analysis.

1. Split the data into training and test set:

```
# Load the data
data("iris")
# Split the data into training (80%) and test set (20%)
set.seed(123)
training.samples <- iris$Species %>%
  createDataPartition(p = 0.8, list = FALSE)
train.data <- iris[training.samples, ]
test.data <- iris[-training.samples, ]
```

2. Normalize the data. Categorical variables are automatically ignored.

```
# Estimate preprocessing parameters
preproc.param <- train.data %>%
  preProcess(method = c("center", "scale"))
# Transform the data using the estimated parameters
train.transformed <- preproc.param %>% predict(train.data)
test.transformed <- preproc.param %>% predict(test.data)
```

Linear discriminant analysis - LDA

The LDA algorithm starts by finding directions that maximize the separation between classes, then use these directions to predict the class of individuals. These directions, called linear discriminants, are a linear combinations of predictor variables.

LDA assumes that predictors are normally distributed (Gaussian distribution) and that the different classes have class-specific means and equal variance/covariance.

Before performing LDA, consider:

- Inspecting the univariate distributions of each variable and make sure that they are normally distribute. If not, you can transform them using log and root for exponential distributions and Box-Cox for skewed distributions.
- removing outliers from your data and standardize the variables to make their scale comparable.

The linear discriminant analysis can be easily computed using the function `lda()` [MASS package].

Quick start R code:

```
library(MASS)
# Fit the model
model <- lda(Species~., data = train.transformed)
# Make predictions
predictions <- model %>% predict(test.transformed)
# Model accuracy
mean(predictions$class==test.transformed$Species)
```

Compute LDA:

```
library(MASS)
model <- lda(Species~., data = train.transformed)
model
```

```
## Call:
## lda(Species ~ ., data = train.transformed)
##
## Prior probabilities of groups:
##     setosa versicolor  virginica
##      0.333      0.333      0.333
##
## Group means:
##            Sepal.Length Sepal.Width Petal.Length Petal.Width
## setosa           -1.012       0.787       -1.293      -1.250
## versicolor        0.117      -0.648        0.272       0.154
## virginica         0.895      -0.139        1.020       1.095
##
## Coefficients of linear discriminants:
##                 LD1     LD2
## Sepal.Length  0.911  0.0318
## Sepal.Width   0.648  0.8985
## Petal.Length -4.082 -2.2272
## Petal.Width  -2.313  2.6544
##
## Proportion of trace:
##    LD1    LD2
## 0.9905 0.0095
```

LDA determines group means and computes, for each individual, the probability of belonging to the different groups. The individual is then affected to the group with the highest probability score.

The `lda()` outputs contain the following elements:

- *Prior probabilities of groups*: the proportion of training observations in each group. For example, there are 31% of the training observations in the setosa group
- *Group means*: group center of gravity. Shows the mean of each variable in each group.
- *Coefficients of linear discriminants*: Shows the linear combination of predictor variables that are used to form the LDA decision rule. for example, `LD1 = 0.91*Sepal.Length + 0.64*Sepal.Width - 4.08*Petal.Length - 2.3*Petal.Width`. Similarly, `LD2 = 0.03*Sepal.Length + 0.89*Sepal.Width - 2.2*Petal.Length - 2.6*Petal.Width`.

Using the function `plot()` produces plots of the linear discriminants, obtained by computing LD1 and LD2 for each of the training observations.

```
plot(model)
```

Make predictions:

```
predictions <- model %>% predict(test.transformed)
names(predictions)
```

```
## [1] "class"     "posterior" "x"
```

The `predict()` function returns the following elements:

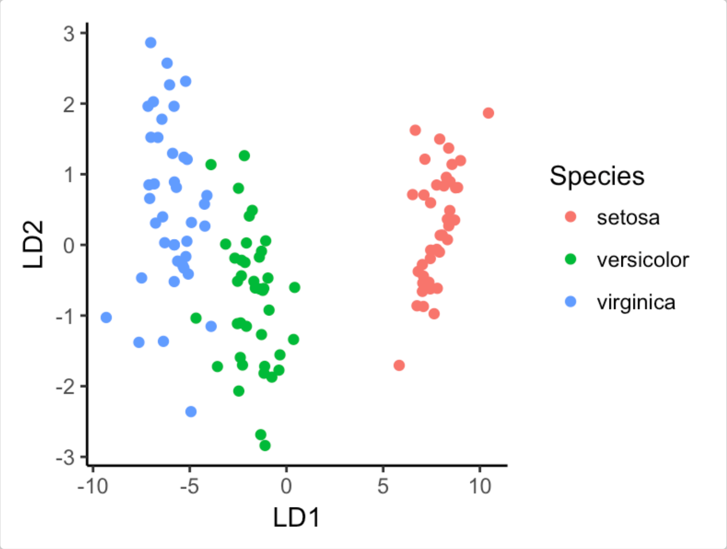
- *class*: predicted classes of observations.
- *posterior*: is a matrix whose columns are the groups, rows are the individuals and values are the posterior probability that the corresponding observation belongs to the groups.
- *x*: contains the linear discriminants, described above

Inspect the results:

```
# Predicted classes
head(predictions$class, 6)
# Predicted probabilities of class memebership.
head(predictions$posterior, 6)
# Linear discriminants
head(predictions$x, 3)
```

Note that, you can create the LDA plot using ggplot2 as follow:

```
lda.data <- cbind(train.transformed, predict(model)$x)
ggplot(lda.data, aes(LD1, LD2)) +
  geom_point(aes(color = Species))
```



Model accuracy:

You can compute the model accuracy as follow:

```
mean(predictions$class==test.transformed$Species)
```

```
## [1] 1
```

✓ It can be seen that, our model correctly classified 100% of observations, which is excellent.

Note that, by default, the probability cutoff used to decide group-membership is 0.5 (random guessing). For example, the number of observations in the setosa group can be re-calculated using:

```
sum(predictions$posterior[ ,1] >=.5)
```

```
## [1] 10
```

In some situations, you might want to increase the precision of the model. In this case you can fine-tune the model by adjusting the posterior probability cutoff. For example, you can increase or lower the cutoff.

Variable selection:

Note that, if the predictor variables are standardized before computing LDA, the discriminator weights can be used as measures of variable importance for feature selection.

Quadratic discriminant analysis - QDA

QDA is little bit more flexible than LDA, in the sense that it does not assumes the equality of variance/covariance. In other words, for QDA the covariance matrix can be different for each class.

LDA tends to be a better than QDA when you have a small training set.

In contrast, QDA is recommended if the training set is very large, so that the variance of the classifier is not a major issue, or if the assumption of a common covariance matrix for the K classes is clearly untenable (James et al. 2014).

QDA can be computed using the R function `qda()` [MASS package]

```
library(MASS)
# Fit the model
model <- qda(Species~., data = train.transformed)
model
# Make predictions
predictions <- model %>% predict(test.transformed)
# Model accuracy
mean(predictions$class == test.transformed$Species)
```

Mixture discriminant analysis - MDA

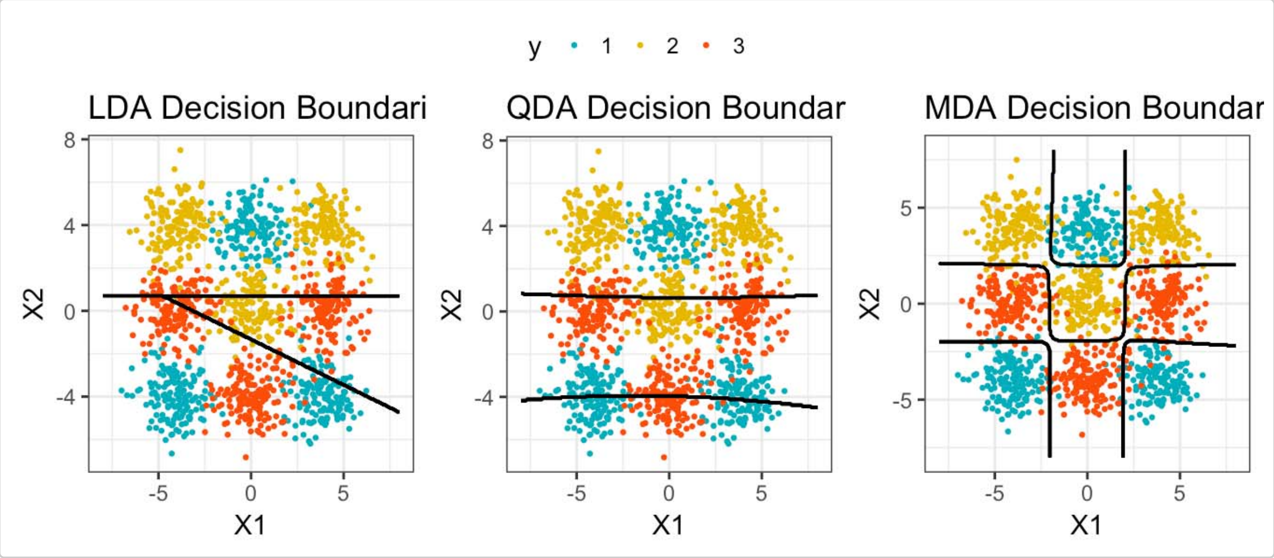
The LDA classifier assumes that each class comes from a single normal (or Gaussian) distribution. This is too restrictive.

For MDA, there are classes, and each class is assumed to be a Gaussian mixture of subclasses, where each data point has a probability of belonging to each class. Equality of covariance matrix, among classes, is still assumed.

```
library(mda)
# Fit the model
```

```
model <- mda(Species~., data = train.transformed)
model
# Make predictions
predicted.classes <- model %>% predict(test.transformed)
# Model accuracy
mean(predicted.classes == test.transformed$Species)
```

MDA might outperform LDA and QDA is some situations, as illustrated below. In this example data, we have 3 main groups of individuals, each having 3 no adjacent subgroups. The solid black lines on the plot represent the decision boundaries of LDA, QDA and MDA. It can be seen that the MDA classifier have identified correctly the subclasses compared to LDA and QDA, which were not good at all in modeling this data.



The code for generating the above plots is from John Ramey

Flexible discriminant analysis - FDA

FDA is a flexible extension of LDA that uses non-linear combinations of predictors such as splines. FDA is useful to model multivariate non-normality or non-linear relationships among variables within each group, allowing for a more accurate classification.

```
library(mda)
# Fit the model
model <- fda(Species~., data = train.transformed)
# Make predictions
predicted.classes <- model %>% predict(test.transformed)
# Model accuracy
mean(predicted.classes == test.transformed$Species)
```

Regularized discriminant analysis

RDA builds a classification rule by regularizing the group covariance matrices (Friedman 1989) allowing a more robust model against multicollinearity in the data. This might be very useful for a large multivariate data set containing highly correlated predictors.

Regularized discriminant analysis is a kind of a trade-off between LDA and QDA. Recall that, in LDA we assume equality of covariance matrix for all of the classes. QDA assumes different covariance matrices for all the classes. Regularized discriminant analysis is an intermediate between LDA and QDA.

RDA shrinks the separate covariances of QDA toward a common covariance as in LDA. This improves the estimate of the covariance matrices in situations where the number of predictors is larger than the number of samples in the training data, potentially leading to an improvement of the model accuracy.

```r
library(klaR)
# Fit the model
model <- rda(Species~., data = train.transformed)
# Make predictions
predictions <- model %>% predict(test.transformed)
# Model accuracy
mean(predictions$class == test.transformed$Species)
```

Discussion

We have described linear discriminant analysis (LDA) and extensions for predicting the class of an observations based on multiple predictor variables. Discriminant analysis is more suitable to multiclass classification problems compared to the logistic regression (Chapter @ref(logistic-regression)).

LDA assumes that the different classes has the same variance or covariance matrix. We have described many extensions of LDA in this chapter. The most popular extension of LDA is the quadratic discriminant analysis (QDA), which is more flexible than LDA in the sens that it does not assume the equality of group covariance matrices.

LDA tends to be better than QDA for small data set. QDA is recommended for large training data set.

References

Friedman, Jerome H. 1989. “Regularized Discriminant Analysis.” *Journal of the American Statistical Association* 84 (405). Taylor & Francis: 165–75. doi:10.1080/01621459.1989.10478752.

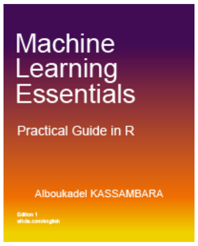
James, Gareth, Daniela Witten, Trevor Hastie, and Robert Tibshirani. 2014. *An Introduction to Statistical Learning: With Applications in R*. Springer Publishing Company, Incorporated.

★★★★★ 1 Note

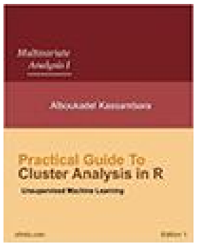
✓ Enjoyed this article? Give us 5 stars ★★★★★ (just above this text block)! Reader needs to be STHDA member for voting. I’d be very grateful if you’d help it spread by emailing it to a friend, or sharing it on Twitter, Facebook or Linked In.

Show me some love with the like buttons below... Thank you and please don't forget to share and comment below!!

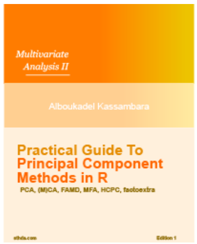
Recommended for You!



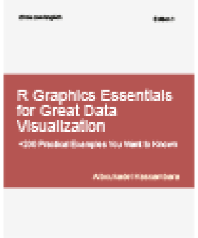
Machine Learning Essentials: Practical Guide in R


Multivariate Analysis I
Alboukadel Kassambara
Practical Guide To Cluster Analysis in R
Unsupervised Machine Learning

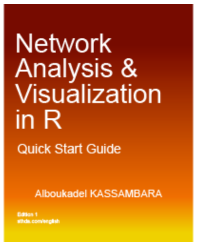
Practical Guide to Cluster Analysis in R


Multivariate Analysis II
Alboukadel Kassambara
Practical Guide To Principal Component Methods in R
PCA, (M)CA, FAMD, MFA, HCPC, factoextra

Practical Guide to Principal Component Methods in R



R Graphics Essentials for Great Data Visualization



Network Analysis and Visualization in R

More books on R and data science

The fields marked with a * are required !

Add a comment

Name

Visitor

Message

Preview

* Code de vérification
How many vowels are in the word sthda?

Submit Reset

SFd 05/02/2018 at 22h52
Visitor

Another super article.
Thank you, STHDA!.

At the end of the **LDA section**,
you say:

"Note that, if the predictor variables are standardized before computing LDA, the discriminator weights can be used as **measures of variable importance** for feature selection".

That is really important and useful.
Can you please elaborate a little on this point?
(and give an example?).

Thanks!
SFfd

#452

kassambara 05/19/2018 at 14h41

Administrator

Feature selection we'll be presented in future blog posts. You can also read the documentation of caret package.

#478

Sign in

Login

Login

Password

Password

Auto connect

☑

Sign in

Register

Forgotten password

Welcome!

Want to Learn More on R Programming and Data Science?

Follow us by Email

Subscribe

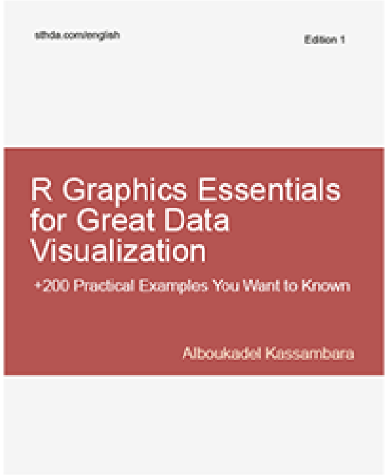
by FeedBurner

on Social Networks

Data Set Example

- factoextra
- survminer
- ggpubr
- ggcorrplot
- fastqcr

Our Books



sthda.com/english

Edition 1

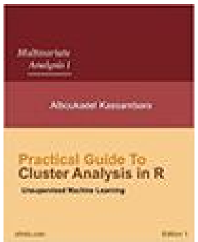
R Graphics Essentials for Great Data Visualization

+200 Practical Examples You Want to Known

Alboukadel Kassambara

R Graphics Essentials for Great Data Visualization: 200 Practical Examples You Want to Know for Data Science

★ NEW!!



Multivariate Analysis I

Alboukadel Kassambara

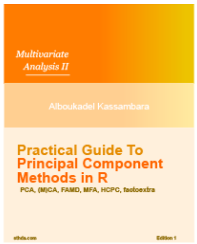
Practical Guide To Cluster Analysis in R

Unsupervised Machine Learning

sthda.com

Edition 1

Practical Guide to Cluster Analysis in R



Multivariate Analysis II

Alboukadel Kassambara

Practical Guide To Principal Component Methods in R

PCA, (M)CA, FAMD, MFA, HCPC, factoextra

sthda.com

Edition 1

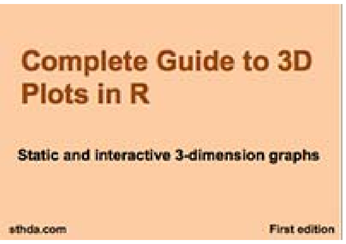
Practical Guide to Principal Component Methods in R

3D Plots in R

STHDA

Data visualization

Alboukadel Kassambara



Complete Guide to 3D Plots in R

Static and interactive 3-dimension graphs

sthda.com

First edition

Guest Book

Many Thanks for the knowledge sharing.
Good luck!

By *Visitor*

Guest Book

R-Bloggers

Newsletter Email

Boosted by PHPBoost

Recommended for you

Cluster Validation Statistics: Must Know...

www.sthda.com

Clustering Distance Measures Essentials...

www.sthda.com

GGPlot Cheat Sheet for Great Customization...

www.sthda.com

AddThis