

1 Introduction

Equivalences between linear discriminant analysis and linear multiple regression.

Linear discriminant analysis and linear regression are both supervised learning techniques. But, the first one is related to classification problems i.e. the target attribute is categorical; the second one is used for regression problems i.e. the target attribute is continuous (numeric).

However, there are strong connections between these approaches when we deal with a binary target attribute. In this particular case, we can even recreate the outputs of the linear discriminant analysis with a linear regression program (Bishop, 2007, pages 189 - 190; Duda et al., 2001, pages 242 – 243; Huberty et Olejnik, 2006, pages 353 – 355; Nakache et Confais, 2003, pages 14 – 16; Saporta, 2006, pages 451 – 452; Tomassone et al., 1988, pages 36 – 38). Unfortunately, if the various references show the connections between the matrix expressions, some explaining the transition formulas, no one details the calculations on a numerical example, making the demonstration too abstract. We perceive badly the real scope of this equivalence. By searching on the Web (in English and French), I ended up finding a detailed example that highlights the relationship. The coefficients of the linear functions from the two approaches are proportional, alas, without that the author details the mathematical expression of the ratio between the coefficients ([Desbois](#), 2003; page 31).

This tutorial takes up the idea. From a practical example, we describe the connections between the two approaches in the case of a binary target variable. We detail the formulas for obtaining the coefficients of discriminant analysis from those of linear regression. It appears that if the equivalence is total when we have balanced dataset i.e. we have the same number of instance for the two classes. In contrast, it is necessary to introduce an additional adjustment of the constant term when the classes are not represented equally ([Hastie et al](#), 2009; page 110). The corresponding formula, not found also in the various references, is detailed.

We perform the calculations under **Tanagra** (balanced data) and **R** (imbalanced data). Our main reference is the book of Tomassone and al. (1988). This book is remarkable on this subject, but also in general on the various themes related to the machine learning problems. Unfortunately, it is not well distributed. No one thought of translating it into English.

2 Dealing with balanced data

2.1 IRIS dataset

We use a modified version of the famous [IRIS dataset](#) in this section. We keep only the two last descriptors {petal-length, petal-width} and $K = 2$ classes {iris-versicolor, iris-virginica}. So, we have $n = 100$ instances (50 + 50). We add also the variable y that we will describe thereafter. Here are the 6 first rows of the dataset (Figure 1).

pet.length	pet.width	species	y
4.7	1.4	versicolor	0.5
4.5	1.5	versicolor	0.5
4.9	1.5	versicolor	0.5
4.0	1.3	versicolor	0.5
4.6	1.5	versicolor	0.5
4.5	1.3	versicolor	0.5

Figure 1 – First rows of the dataset – Binary IRIS

Since we have 2 descriptors, we can plot the data points in a scatterplot. We differentiate the instances according to their class membership.

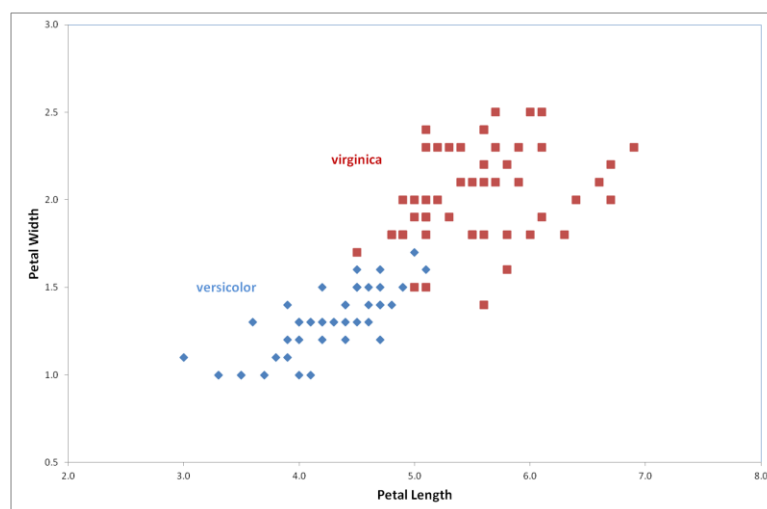


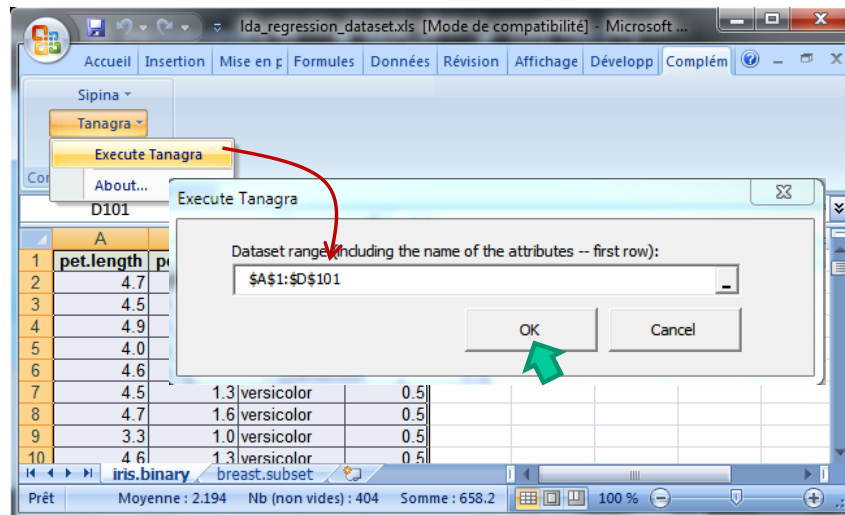
Figure 2 – Scatterplot – Class membership

The two groups of individuals are rather distinct. Finding a linear boundary that allows to separate them will be easy. The error rate of the model should be low. Misclassified individuals will be located in the overlapping parts of the conditional point clouds.

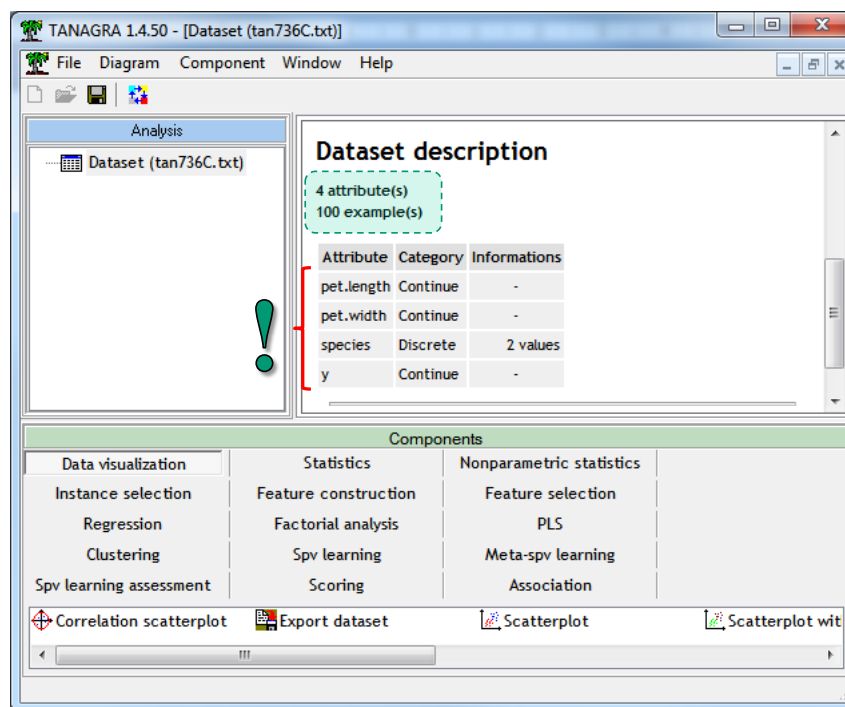
2.2 Linear discriminant analysis with Tanagra – Reading the results

2.2.1 Data importation

We want to perform a linear discriminant analysis with Tanagra. We open the "lda_regression_dataset.xls" file into Excel, we select the whole data range and we send it to Tanagra using the "[tanagra.xla](#)" add-in.

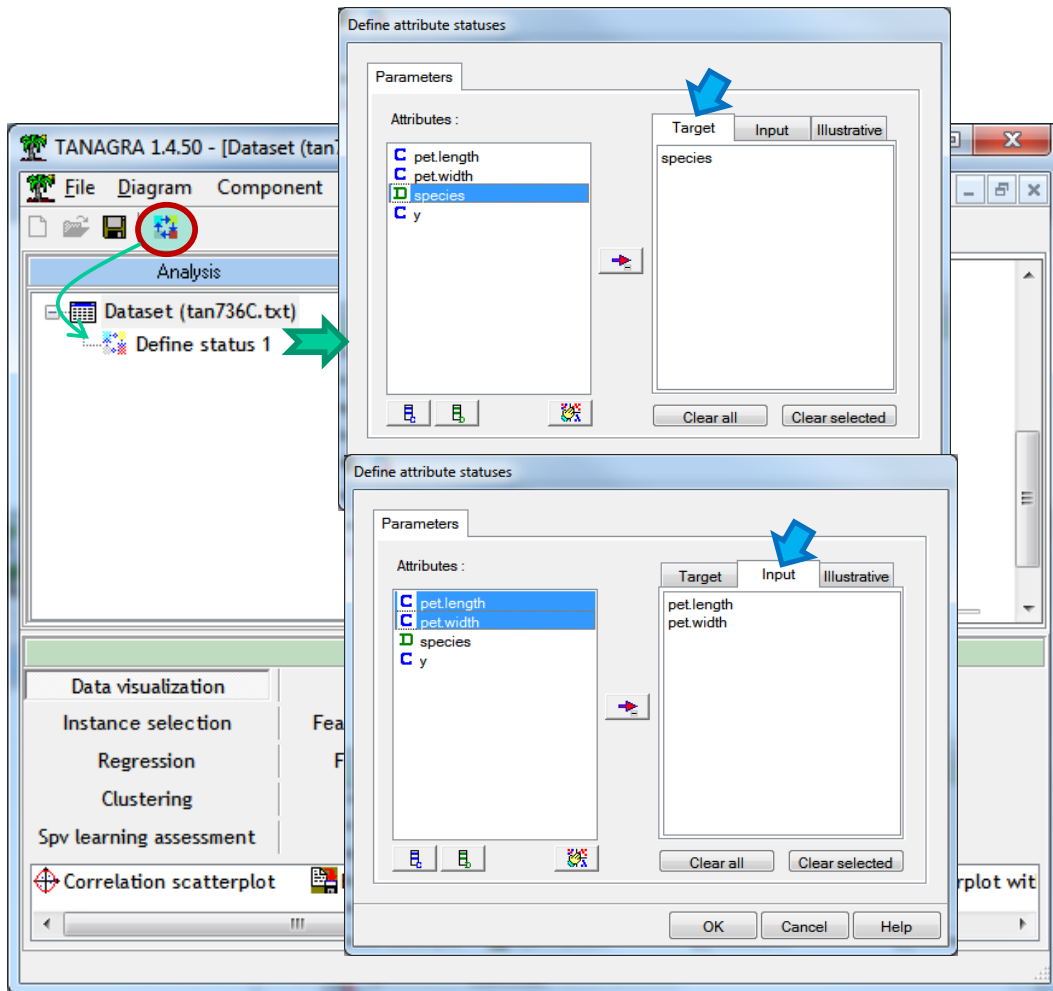


Tanagra is automatically launched; 4 columns are imported with 100 instances.

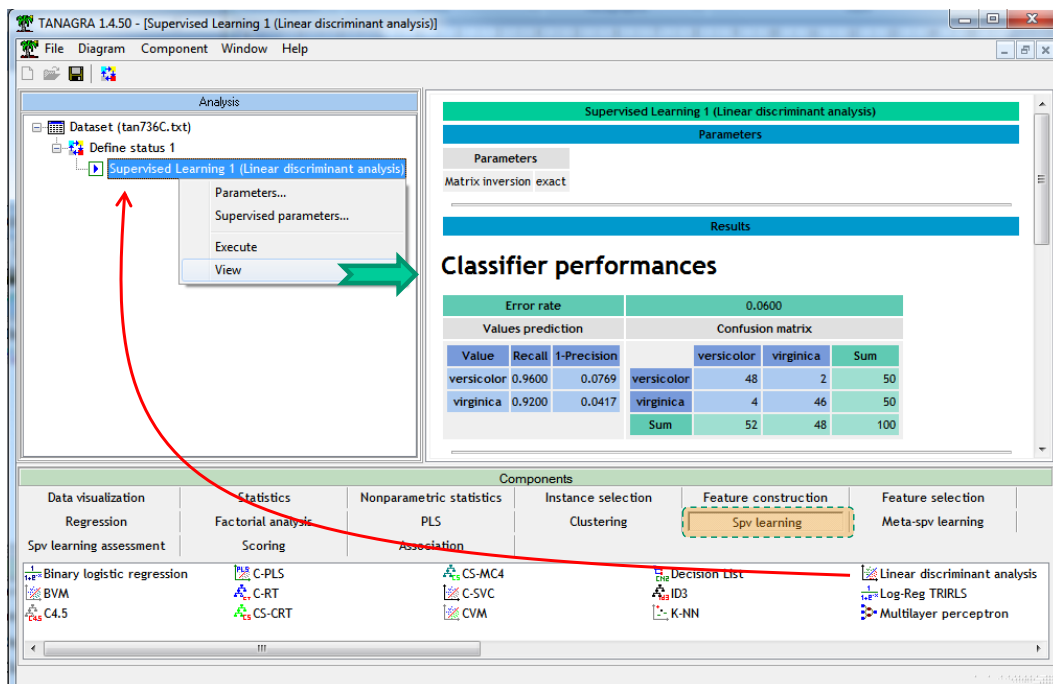


2.2.2 Discriminant analysis

First, we must define the status of the variables. We use the DEFINE STATUS component for that. We click on the shortcut into the toolbar. We set SEPCIES as **target**, PET.LENGTH and PET.WIDTH as **input**. The variable Y is not used at this stage.



We add the LINEAR DISCRIMINANT ANALYSIS (SPV LEARNING tab) into the diagram.



We click on the VIEW contextual menu to obtain the results.

2.2.3 Reading the results

Confusion matrix. The “Classifier performances” part incorporates the confusion matrix computed on the learning sample.

Classifier performances

Error rate			0.06			
Values prediction			Confusion matrix			
Value	Recall	1-Precision		versicolor	virginica	Sum
versicolor	0.96	0.0769	versicolor	48	2	50
virginica	0.92	0.0417	virginica	4	46	50
			Sum	52	48	100

Figure 3 – Confusion matrix

We have a balanced dataset with $n_1 = 50$ « G_1 : versicolor » and $n_2 = 50$ « G_2 : virginica ». 6 instances are misclassified (error rate: $6 / 100 = 6\%$), with 4 instances “virginica” labeled “versicolor”, and 2 conversely. We can visualize them when we draw the boundary separating the classes in the representation space (Figure 7).

MANOVA. The multivariate analysis of variance corresponds to a test for comparison of conditional centroids. The Wilks’ lambda (Λ) is the ratio between the within-group variance and the total variance. The closer it gets to 0, the furthest are the conditional centroids. For our dataset, we have $\Lambda = 0.2802$. This suggests a good separation of the groups, confirmed on the one hand by the scatterplot of conditional data points (Figure 2), on the other hand, by the low error rate (Figure 3).

MANOVA

Stat	Value	p-value
Wilks' Lambda	0.2802	-
Bartlett -- C(2)	123.3935	0
Rao -- F(2, 97)	124.5641	0

Figure 4 – MANOVA test

The Wilks' lambda can be applied to any number of classes ($K \geq 2$). For the binary problem ($K = 2$), we can compute the distance between the centroids μ_1 (versicolor) and μ_2 (virginica). We use the “Mahalanobis distance” (D), it is defined as follows:

$$D^2 = \frac{1 - \Lambda}{\Lambda} \times \frac{n(n - 2)}{n_1 \times n_2}$$

For our dataset,

$$D^2 = \frac{1 - 0.2802}{0.2802} \times \frac{100(98)}{50 \times 50} = 10.0678$$

We can visualize the centroids (μ_1, μ_2) – with the coordinates $\mu_1 = (4.26, 1.33)$ and $\mu_2 = (5.55, 2.03)$ – and their distance D^2 (Figure 5).

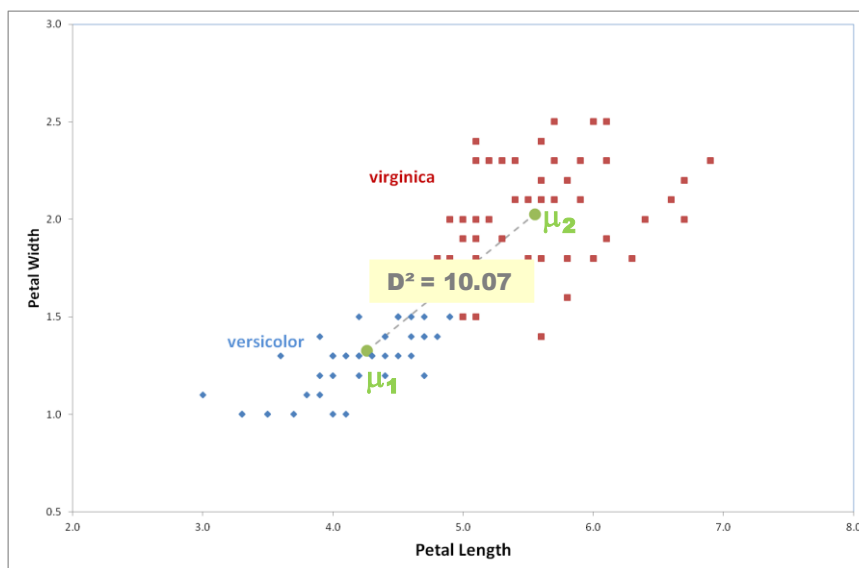


Figure 5 – Conditional centroids - Mahalanobis distance (D^2)

To test the significance of the difference, we use the Rao's F statistic which follows a Fisher distribution under the null hypothesis (the centroids are identical). For our dataset, we have $F = 124.5641$, the statistic follows a Fisher distribution at (2, 97) degrees of freedom. We note that we reject the null hypothesis at the 5% level (Figure 4).

Classification functions – Score function. The classification functions can be used to determine to which group each instance most likely belongs. There are as many classification functions as there are groups (Huberty and Olejnik, 2006; page 274).

$$D(G_1, X) = a_0 + a_1 * X_1 + a_2 * X_2$$

$$D(G_2, X) = b_0 + b_1 * X_1 + b_2 * X_2$$

In the binary problems ($K = 2$), we can compute a linear "score" function¹ which is formed from the difference term by term of the coefficients provided by the classification functions. Applied to an instance, it returns a value which is proportional to the level of membership to the group G_1 . It is an alternative to the LOGIT function provided by the logistic regression.

$$D(X) = \theta_0 + \theta_1 * X_1 + \theta_2 * X_2$$

With

$$\theta_j = (a_j - b_j)$$

¹ "Score" function is maybe not the best way to designate it in English. But it corresponds to the usual practice in the French-speaking world.

Tanagra provides the classification functions, we can infer the "score" function.

	Classification functions		Score function
Attribute	versicolor	virginica	D(X)
pet.length	14.40029	17.164859	-2.764569
pet.width	7.824622	17.104674	-9.280052
constant	-36.55349	-65.66983	29.116340

Figure 6 – Classification functions and score function

The classification rule for an unseen instance ω is:

IF $D[X(\omega)] \geq 0$ **THEN** Versicolor **ELSE** Virginia

Thus, for an instance with the following values (pet.length = 4.7, pet.width = 1.4):

$$D = 29.116340 + (-2.764569 * 4.7) + (-9.280052 * 1.4) = 3.13 > 0$$

The class "versicolor" is assigned to the instance. This seems obvious when we consider the location of the instance into the representation space (Figure 7).

Boundary between classes. $D(X) = 0$ defines the boundary allowing to separate the classes into the representation space. In the two-dimensional representation space, it corresponds to a straight line (Figure 7).

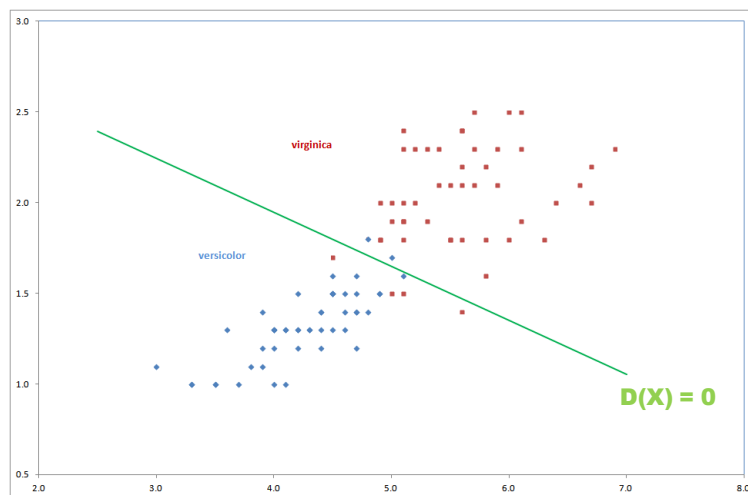


Figure 7 - Boundary defined by the linear discriminant analysis

We observe the 6 misclassified instances in either side of the boundary (Figure 7). These are those that highlighted in the confusion matrix (Figure 3).

Relevance of the predictive variables. The "Statistical Evaluation" part of the coefficients table enables to appreciate the variable importance in the model. One possible point of view is that it is based on a statistical test allowing to check if the coefficients of a variable are identical whatever the classification function.

Concretely, the test statistic F_j is based on the comparison of the Wilks' lambda Λ with and without the variable X_j to evaluate. Under the null hypothesis, it follows a Fisher distribution with $(1, n - p - K + 1)$ degrees of freedom $[(1, n - p - 1)$ since $K = 2$ for our dataset].

LDA Summary

Attribute	Classification functions		Statistical Evaluation			
	versicolor	virginica	Wilks L.	Partial L.	F(1,97)	p-value
pet.length	14.40029	17.164859	0.314202	0.89192	11.75412	0.000893
pet.width	7.824622	17.104674	0.381538	0.734509	35.06098	0.000000
constant	-36.55349	-65.66983	-			

Figure 8 – Relevance of the input variables – Linear discriminant analysis

We note that the two variables are both relevant (significant) at the 5% level. In particular, we will remember the values of F to compare them with the significance test statistics of the linear regression below.

2.3 Comparison with SAS

The same results are available with two procedures of the [SAS software](#). The [PROC DISCRIM](#) provides the global evaluation and the classification functions.

```
proc discrim data = mesdata.iris_binary manova;
class species;
var pet_length pet_width;
priors proportional;
run;
```

We obtain (see Figure 4 and Figure 6).

The SAS System

The DISCRIM Procedure

Multivariate Statistics and Exact F Statistics					
S=1 M=0 N=47.5					
Statistic	Value	F Value	Num DF	Den DF	Pr > F
Wilks' Lambda	0.28024304	124.56	2	97	<.0001
Pillai's Trace	0.71975696	124.56	2	97	<.0001
Hotelling-Lawley Trace	2.56833127	124.56	2	97	<.0001
Roy's Greatest Root	2.56833127	124.56	2	97	<.0001

Linear Discriminant Function for species			
Variable	Label	versicolor	virginica
Constant		-36.55349	-65.66983
pet_length	pet#length	14.40029	17.16486
pet_width	pet#width	7.82462	17.10467

The PROC STEPDISC provides the test statistic F_j allowing to measure the variable importance.

```
proc stepdisc data = mesdata.iris_binary method = backward;
class species;
var pet_length pet_width;
run;
```

The statistic F are identical to those of Tanagra (Figure 8).

The SAS System

The STEPDISC Procedure
Backward Elimination: Step 1

Statistics for Removal, DF = 1, 97				
Variable	Label	Partial R-Square	F Value	Pr > F
pet_length	pet#length	0.1081	11.75	0.0009
pet_width	pet#width	0.2655	35.06	<.0001

2.4 Linear regression for the classification process

2.4.1 Principle – Working with a coded target attribute

The aim of the [linear regression](#) is to explain (predict) the values of a numeric dependent variable with one or more independent variables. We dispose of many tools to evaluate the model in its globality and the influence of each independent variables.

Let us see how to perform a linear regression on our modified version of the IRIS dataset. We must code appropriately the categorical target attribute SPECIES which takes two values $\{G_1: \text{versicolor}, G_2: \text{virginica}\}$. The coded target attribute Y takes two possible numerical values (y_1, y_2) , which are defined as follows for an individual ω :

$$y(\omega) = \begin{cases} y_1 & \text{when } \omega \in G_1 \\ y_2 & \text{when } \omega \in G_2 \end{cases}$$

We obtain a regression equation:

$$R(X) = \beta_0 + \beta_1 * X_1 + \beta_2 * X_2$$

Where β_j are the coefficients of the model.

For an unseen instance to classify ω , the classification rule is:

IF $R[X(\omega)] \geq \bar{y}$ **THEN** Versicolor **ELSE** Virginia

The threshold value \bar{y} is the average of the variable Y

$$\bar{y} = \frac{n_1 \times y_1 + n_2 \times y_2}{n}$$

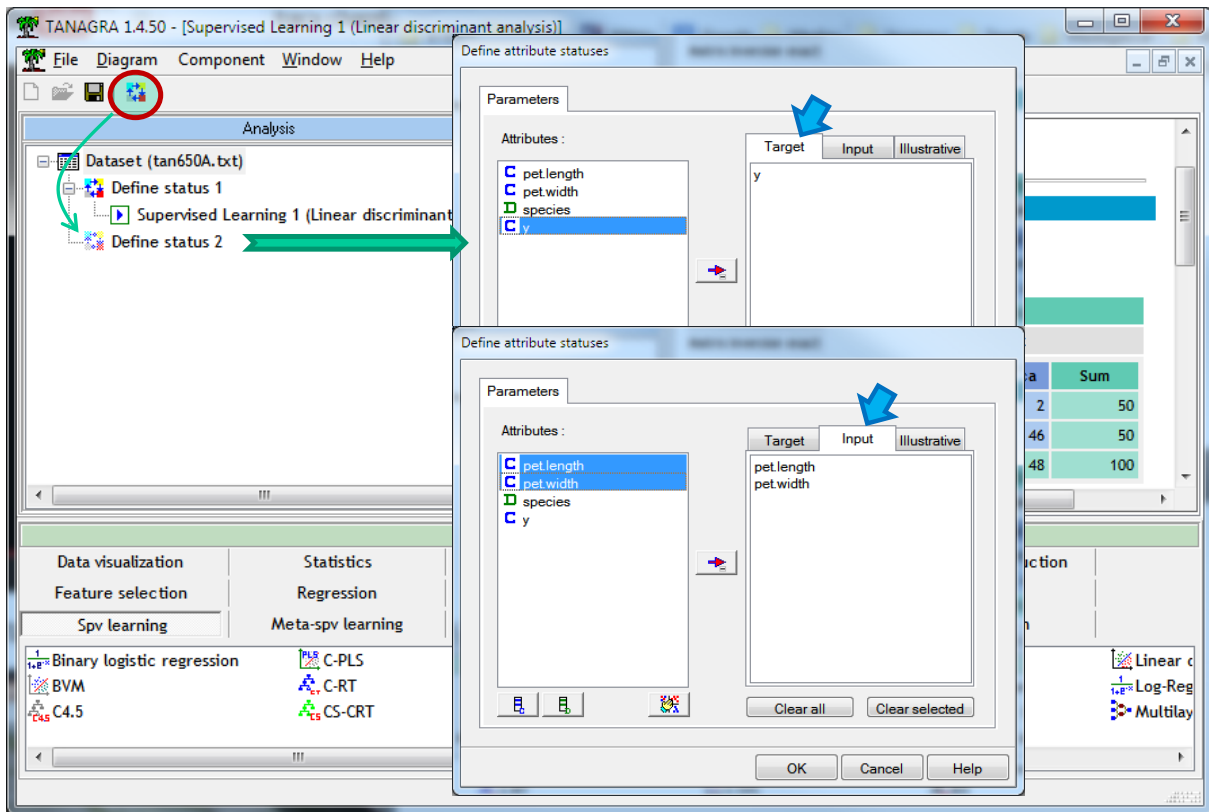
Coding values. Any coding values are adapted as long as $y_1 \neq y_2$. Several options are possible.

- The simplest: ($y_1 = 1 ; y_2 = 0$). In this case, the threshold value is $\bar{y} = \frac{n_1}{n}$. De facto, the threshold 0.5 is a particular situation which is adapted when we have balanced dataset ($n_1 = n_2$).
- The coding values $\left(y_1 = \frac{n_2}{n}; y_2 = -\frac{n_1}{n} \right)$ (Tomassone, 1988; page 38) have the advantage to infer a null threshold for the reason that $\bar{y} = 0$. The regression equation is similar to a score function in this case. We will see that they are fully equivalent when $n_1 = n_2$.
- Other coding values which leads to a null threshold ($\bar{y} = 0$) are possible:
 - $\left(y_1 = \frac{n}{n_1}; y_2 = -\frac{n}{n_2} \right)$ (Duda and al., 2001, page 242; Saporta, 2006, page 451);
 - $\left(y_1 = \sqrt{\frac{n_2}{n_1}}; y_2 = -\sqrt{\frac{n_1}{n_2}} \right)$ (Nakache and Confais, 2003; page 14); etc.

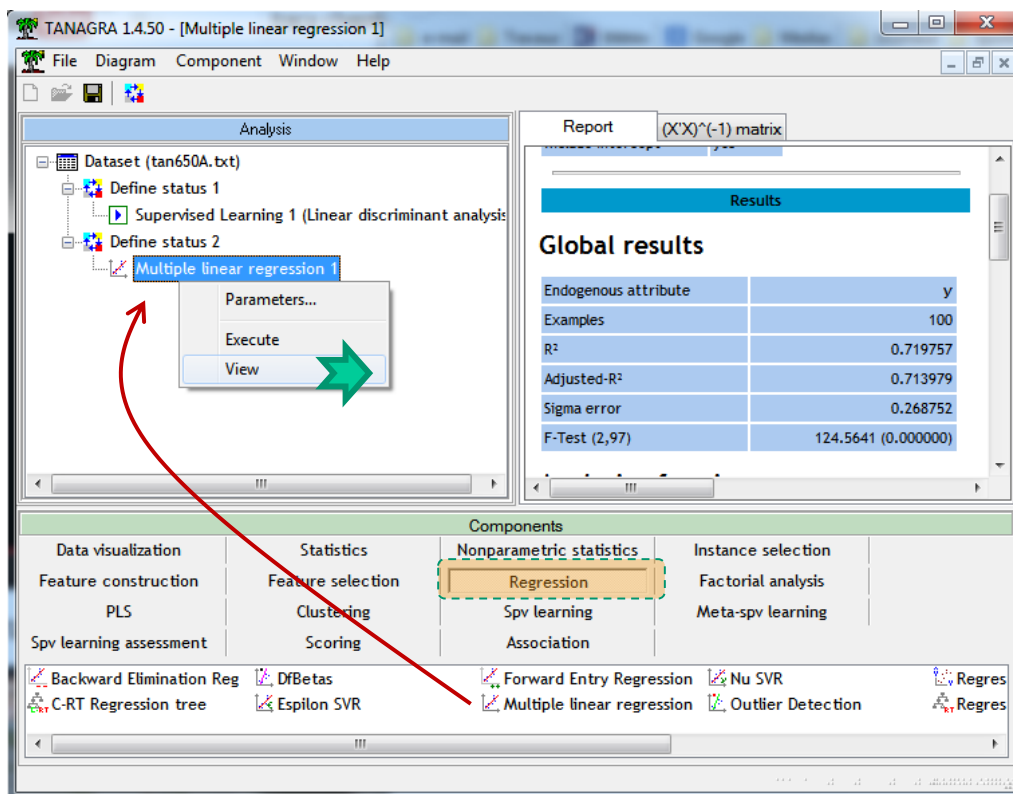
We choose the coding values $\left(y_1 = \frac{n_2}{n} = 0.5; y_2 = -\frac{n_1}{n} = -0.5 \right)$ for our dataset (variable **y**). It corresponds to the last column of our data file (Figure 1).

2.4.2 Multiple linear regression with Tanagra

We go back to Tanagra. We add a new DEFINE STATUS component into the diagram. We set Y as TARGET, PET.LENGTH and PET.WIDTH as INPUT.



We insert the tool MULTIPLE LINEAR REGRESSION (REGRESSION tab). We click on the VIEW contextual menu to visualize the results.



Let us see the details.

2.4.3 Overall model fit

The R-Square (R^2) is the main tool for the evaluation of the model. This is the proportion of variance explained by the model. For our dataset, we have $R^2 = 0.719757$.

Global results

Endogenous attribute	y
Examples	100
R^2	0.719757
Adjusted-R	0.713979
Sigma error	0.268752
F-Test (2,97)	124.5641 (0.000000)

Figure 9 – Overall model fit

We can associate to the R^2 the test statistic F. It enables to test the global significance of the model (H_0 : all the coefficients associated to the variables are equal to 0). Under H_0 , it follows a Fisher distribution with $(p, n - p - 1)$ degrees of freedom. We obtain $F = 124.5641$; the model is globally significant at the 5% level (Figure 9).

2.4.4 Regression coefficients and tests for significance

This table provides the estimated coefficients of the model β_j (**Coef.**). The column "**t(97)**" is the t-statistic t_j for significance ($H_0 : \beta_j = 0$). It follows a Student distribution at $(n - p - 1)$ degrees of freedom.

Coefficients

Attribute	Coef.	std	t(97)	p-value
pet.length	-0.197641	0.057648	-3.428428	0.000893
pet.width	-0.663436	0.112044	-5.921231	0.000000
Intercept	2.081544	0.168871	12.326226	0.000000

Figure 10 – Regression Coefficients - Tests for significance

A quick comparison allows to observe that the ratio between the coefficients of the score function from the linear discriminant analysis (LDA, Figure 6) and the regression equation (REG, Figure 10) is the same whatever the variable being considered, including the constant:

$$\frac{-2.764569}{-0.197641} = \frac{-9.280052}{-0.663436} = \frac{29.11634}{2.081544} = 13.98$$

This phenomenon has also been noticed on another part of the IRIS dataset [setosa vs. versicolor] (Desbois, 2003; page 31).


	Score function	Coefficients	
	LDA	REG	 Ratio
pet. length	-2.764569	-0.197641	13.98783
pet. width	-9.280052	-0.663436	13.98786
constant	29.116340	2.081544	13.98786

Figure 11 - Ratio between the coefficients of the score function (LDA) and the regression (REG)

Therefore, the **linear regression** for the classification as we define it in this section provides a result fully equivalent to that of the **linear discriminant analysis**. Both approaches construct the **same boundary line** to separate the classes.

2.5 Transition formula and equivalences

Observing the equivalence retrospectively is a good thing. But the real issue is to be able to calculate this ratio a priori, in order to deduce the results of linear discriminant analysis (LDA) from the linear regression (REG). This is what we show in this section.

2.5.1 From R^2 to Λ - Equivalence between the global evaluation of the models

The R^2 (R-squared) of the regression is obtained from the ratio between the explained variance and the total variance. The Wilks' lambda (Λ) of the linear discriminant analysis is the ratio between the residual variance (within-group variance) and the total variance. The following relation comes naturally:

$$\Lambda = 1 - R^2 = 1 - 0.719757 = 0.280243$$

We find the result of the LDA. The tests for global significance are identical with $F = 124.5641$ which follows a Fisher distribution at (2, 97) degrees of freedom (Figure 4 and Figure 9).

2.5.2 Transition formula between the coefficients

Since we have Λ , we can calculate the Mahalanobis distance D between the centroids. We obtain $D^2 = 10.0678$ (see page 5).

To simplify the expressions, we set:

$$c_1 = n_1 + n_2 - 2 = n - 2 = 100 - 2 = 98$$

And

$$c_2 = \frac{n_1 \times n_2}{n_1 + n_2} = \frac{50 \times 50}{50 + 50} = 25$$

We obtain the ratio between the coefficients of LDA and REG with (Tomassone and al., 1988):

$$\rho = \frac{\theta_j}{\beta_j} = \frac{c_1 + c_2 \times D^2}{c_2 \times (y_1 - y_2)} \quad (j = 0, 1, \dots, p)$$

For the IRIS dataset, we perform the following calculation:

$$\rho = \frac{98 + 25 \times 10.0678}{25 \times (0.5 - (-0.5))} = 13.98786$$

This is the value obtained when we calculate retrospectively the ratio between the coefficients of LDA and REG (Figure 11). This ratio ρ is the same whatever the coefficients, including the constant term when we have balanced dataset ($n_1 = n_2$).

2.5.3 Tests for significance of coefficients

For the regression, we have the t-statistic t_j which follows a Student distribution with $(n - p - 1)$ degrees of freedom. For the discriminant analysis, we have F_j which follows a Fisher distribution with $(1, n - p - 1)$ degrees of freedom. The following relation is obvious:

$$F_j = t_j^2$$

For instance, for the first explanatory variable (PET.LENGTH) (Figure 10 and Figure 8), we have:

$$F_1 = t_1^2 = (-3.428428)^2 = 11.75412$$

Here also, we can directly use the results of the regression to measure the relevance of the variables in the binary linear discriminant analysis.

3 Handling imbalanced dataset

The regression provides a constant term which is not proportional to the one of the score function of LDA when we deal with imbalanced dataset ($n_1 \neq n_2$). The boundary provided by the regression is parallel to the one of the discriminant analysis. Therefore, the regression model has not the same behavior than the linear discriminant model since the classification rule is different ([Hastie et al, 2009; page 110](#)). [An additional correction must be introduced for the constant term to obtain the equivalence.](#)

3.1 Additional correction for the constant term

The relation between the coefficients of the variables remains the same: $\theta_j = \rho \times \beta_j$ ($j \geq 1$).

An additional correction δ is need for the constant term:

$$\tilde{\theta}_0 = \theta_0 + \delta = \rho \times \beta_0 + \delta$$

The correction δ is based on the distribution of classes and the coordinates of the centroids. It can be obtained from the coefficients related to the independent variables from the score function (Nakache and Confais, 2003, page 19; the authors describes the [Fisher's discriminant function](#) and, consequently, omit the part relating to the groups sample sizes n_1 and n_2) :

$$\delta = \ln \frac{n_1}{n_2} - \frac{1}{2} \sum_{j=1}^p \theta_j \times [(\mu_1^j + \mu_2^j) - 2 \times \mu^j]$$

Where μ^j is the mean of the variable X_j for all the instances, μ_1^j (resp. μ_2^j) the mean of the variable X_j for the instances from the group G_1 (resp. G_2).

Note: We observe that $\delta = 0$ when we have balanced dataset ($n_1 = n_2$). Indeed, in this case:

$$\ln \left(\frac{n_1}{n_2} \right) = \ln(1) = 0$$

And,

$$\mu^j = \frac{n_1 \times \mu_1^j + n_2 \times \mu_2^j}{n} = \frac{1}{2} (\mu_1^j + \mu_2^j) \Rightarrow (\mu_1^j + \mu_2^j) - 2\mu^j = 0$$

3.2 BREAST dataset

To illustrate the calculations for imbalanced dataset, we use a part of the well-known "breast-cancer-wisconsin"² dataset, with only $p = 3$ descriptors (clump, ucellsize, ucellshape). The target attribute TARGET is binary³ ($K = 2$). The first G_1 is the class "benign", G_2 corresponds to "malignant". We have $n = 699$ instances, with $n_1 = 458$ and $n_2 = 241$. Here are the first rows of the dataset.

clump	ucellsize	ucellshape	target
4	2	2	benign
1	1	1	benign
2	1	1	benign
10	6	6	malignant
4	1	1	benign

3.2.1 Coding the target variable

The first step consists in coding the target attribute, we create Y with two possible values:

$$y_1 = \frac{n_2}{n} = \frac{241}{699} = 0.345 \quad \text{and} \quad y_2 = -\frac{n_1}{n} = -\frac{458}{699} = -0.655$$

3.2.2 Results of the regression

We send Y and the $p = 3$ independent variables to Tanagra. We perform the regression analysis. We obtain the following results.

² <http://archive.ics.uci.edu/ml/datasets/Breast+Cancer+Wisconsin+%28Original%29>

³ We changed the name of the variable "class" to "target" to avoid confusion when processing under R below.

Global results

Endogenous attribute	y
Examples	699
R ²	0.747486
Adjusted-R ²	0.746396
Sigma error	0.239526
F-Test (3,695)	685.7753 (0.000000)

Coefficients

Attribute	Coef.	std	t(695)	p-value
clump	-0.048006	0.004315	-11.124401	0
ucellsize	-0.053245	0.007144	-7.453079	0
ucellshape	-0.051713	0.007415	-6.973756	0
Intercept	0.544840	0.016888	32.262169	0

At this stage, we have all the elements to calculate the ratio ρ between the coefficients of the regression and the score function of the linear discriminant analysis.

3.2.3 Calculating the ratio ρ - Calculation the coefficients of the score function

Several steps are needed to achieve this. We must first calculate the Wilks' lambda (Λ) from the R-squared (R^2) of the regression:

$$\Lambda = 1 - R^2 = 1 - 0.747486 = 0.252514$$

Then, we calculate the Mahalanobis distance:

$$D^2 = \frac{1 - \Lambda}{\Lambda} \times \frac{n(n-2)}{n_1 \times n_2} = \frac{1 - 0.252514}{0.252514} \times \frac{699(699-2)}{458 \times 241} = 13.06607$$

We calculate c_1 and c_2 to be consistent with the presentation of the previous section:

$$c_1 = n_1 + n_2 - 2 = n - 2 = 699 - 2 = 697$$

$$c_2 = \frac{n_1 \times n_2}{n_1 + n_2} = \frac{458 \times 241}{458 + 241} = 157.908$$

We finally get ρ

$$\rho = \frac{c_1 + c_2 \times D^2}{c_2 \times (y_1 - y_2)} = \frac{697 + 157.908 \times 13.06607}{157.908 \times (0.345 - (-0.655))} = 17.48002$$

Thus, from the coefficients of the regression β_j , we can compute the coefficients of the score function $\theta_j = \beta_j \times \rho$:

	Beta_j	Theta_j
clump	-0.048006	-0.83915
ucellsize	-0.053245	-0.93072
ucellshape	-0.051713	-0.90394
Intercept	0.544840	9.52382

3.2.4 Correction of the constant term (δ)

To adjust the constant term, we must calculate the centroids (overall and conditional).

Classes	Barycentres		
	_clump	_ucellsize	_ucellshape
mu_1	2.956	1.325	1.443
mu_2	7.195	6.573	6.560
mu	4.418	3.134	3.207

Then, we calculate δ :

$$\delta = \ln \frac{458}{241} - \frac{1}{2} \{-0.83915 \times [(2.956 + 7.195) - 2 \times 4.418] + \dots\} = 2.67021$$

Thus, the adjusted constant term is:

$$\tilde{\theta}_0 = \theta_0 + \delta = 9.52382 + 2.67021 = 12.19403$$

Now, we have all the coefficients of the LDA score function:

	Score function LDA by REG
clump	-0.83915
ucellsize	-0.93072
ucellshape	-0.90394
Intercept	12.19403


3.2.5 Comparison with the LDA score function of Tanagra

When we perform directly the LDA with Tanagra, we obtain coefficients (Figure 12) which are consistent with those obtained from the post processing of the linear regression coefficients. The small differences are due to truncation errors in the intermediate calculations.

MANOVA

Stat	Value	p-value
Wilks' Lambda	0.2525	-
Bartlett -- C(3)	957.2095	0
Rao -- F(3, 695)	685.7753	0

LDA Summary



Attribute	Classification		Score	Statistical Evaluation			
	begin	malignant	Function	Wilks L.	Partial L.	F(1,695)	p-value
clump	0.70839	1.54754	-0.83915	0.297477	0.848853	123.75231	0
ucellsize	0.13147	1.06218	-0.93072	0.272696	0.92599	55.54839	0
ucellshape	0.25922	1.16318	-0.90395	0.270184	0.9346	48.63328	0
constant	-1.74408	-13.93812	12.19404	-			

Figure 12 – LDA results - "Breast" dataset

3.3 An example of processing under R

In order for the reader to be able to easily reproduce the process and, why not, to transpose it to other files, I propose to resume the whole procedure as a R program in this section. Here is the commented source code.

```
#data importation
library(xlsx)
breast <- read.xlsx(file="lda_regression_dataset.xls", header=T, sheetIndex=2)
print(summary(breast))

#sample sizes
n1 <- table(breast$target)[1] #begin
n2 <- table(breast$target)[2] #malignant
n <- n1+n2

#coding the target attribute - Tomassone, page 38
y1 <- n2/n
y2 <- -n1/n
y <- ifelse(breast$target=="begin",y1,y2)

#regression on the coded target attribute
reg <- lm(y ~ ., data = breast[-4])
print(reg)
```

```
beta <- reg$coefficients
print(round(beta,5))

#summary
sreg <- summary(reg)

#R2 (R-squared) of the regression
R2 <- sreg$r.squared

#D2 (Mahalanobis distance) - Huberty, page 353; Tomassone, page 38
D2 <- (R2/(1-R2))*(n*(n-2))/(n1*n2)
names(D2)[1] <- "D2"
print(D2)

#intermediate results for the calculations (Tomassone, page 27)
c1 <- n1+n2-2
c2 <- (n1*n2)/(n1+n2)

#rho - correction factor
rho <- (c1+c2*D2)/(c2*(y1-y2))
print(rho)

# score function before the adjustment of the constant term
theta <- beta*rho
print(round(theta,5))

*** correction of the constant term ***

#1st adjustment
e1 <- log(n1/n2)

#average
mu <- sapply(breast[1:3],mean)

#conditional average
mu.cond <- aggregate(breast[1:3],by=list(breast$target),mean)[2:4]

#adjustment on the averages
mu.centre <- ((mu.cond[1,]+mu.cond[2,])-2*mu)

#coef. Of the LDA (without the constant term)
```

```
coef.lda.p <- theta[2:4]

#scalar product – 2nd correction
e2 <- -0.5*sum(coef.lda.p*mu.centre)

#delta
delta <- e1 + e2
print(delta)

#correction of the constant term
theta_tilde <- theta
theta_tilde[1] <- theta[1] + delta

#LDA score function after all the adjustments
print(round(theta_tilde,5))

*** comparaison des performances ***

#confusion matrix and error rate
confusion.matrix <- function(dataset,coef){
  #prediction for one row
  prediction <- function(data.row){
    score <- sum(data.row[1:3]*coef[2:4])+coef[1]
    return(ifelse(score>=0,"benign","malignant"))
  }
  #prediction for all rows
  pred <- factor(apply(data.matrix(dataset),1,prediction))
  #confusion matrix
  cm <- table(dataset$target,pred)
  print(cm)
  #error rate
  er <- 1-sum(diag(cm))/sum(cm)
  print(er)
}

#confusion matrix - regression
confusion.matrix(breast,beta)

#confusion matrix - lda
confusion.matrix(breast,theta_tilde)
```

Here are the main outputs of the program.

Coefficients of the regression β .

```
> print(round(beta,5))
(Intercept)      clump  ucellsize  ucellshape
      0.54484    -0.04801    -0.05324    -0.05171
```

Mahalanobis distance (D^2) obtained from the R-squared (R^2) of the regression.

```
> print(D2)
      D2
13.06609
```

Calculation of the ratio ρ .

```
> print(rho)
      rho
17.48004
```

1st version of the score function

```
> print(round(theta,5))
(Intercept)      clump  ucellsize  ucellshape
      9.52382    -0.83915    -0.93072    -0.90395
```

Correction δ for the constant term

```
> print(delta)
      delta
2.670214
```

Score function after adjustment of the constant (see Figure 12).

```
> print(round(theta_tilde,5))
(Intercept)      clump  ucellsize  ucellshape
      12.19404    -0.83915    -0.93072    -0.90395
```

Comparison of the accuracy

```
> #matrice de confusion regression
> confusion.matrix(breast,beta)
      pred
      begin malignant
begin      435      23
malignant   9      232
[1] 0.04577969
>
> #matrice de confusion lda
> confusion.matrix(breast,theta_tilde)
      pred
      begin malignant
begin      448      10
malignant  33      208
[1] 0.06151645
```

Strangely, the regression (Error rate = 4.58%) would be more efficient than the discriminant analysis (Error rate = 6.15%) on our dataset. But, before leaping to any conclusions, we must note that this is only an example on a single dataset. Moreover, the performance is evaluated in resubstitution i.e. we use the same sample for learning and testing phases. It just confirms that the regression (before correction of the constant) and discriminant analysis produce different models when the classes are unbalanced. That explains the disparities between the confusion matrices.

4 Conclusion

Working on this tutorial was particularly exciting. I knew for a long time that it was possible to get the results of the binary discriminant analysis from multiple linear regression since everyone was talking about it. But it is quite different to detail the process when we must explain it in a tutorial. It was necessary to identify the correct transition formula at each step, and rebuild the expression to rectify the constant when the classes are not balanced. Surprisingly, this process is not really well described in the various books I have consulted.

The synonymy between these two approaches exceeds the scientific curiosity. It legitimizes the use of a multiple regression program/algorithm to perform a binary linear discriminant analysis. The results are fully equivalent, but the processing will be faster because the calculations and data structures are simpler for regression, increasing our ability to process large databases. This advantage is even more important in a variable selection process which requires more computing resources.

5 References

- C.M. Bishop, « Pattern Recognition and Machine Learning », Springer, 2007.
- D. Desbois, « [Une introduction à l'analyse discriminante avec SPSS pour Windows](#) », Revue Modulad, n°30, 2003.
- R.O. Duda, P.E. Hart, D. Stork, « Pattern Classification », 2nd Edition, Wiley, 2000.
- T. Hastie, R. Tibshirani, J. Friedman, « [The Elements of Statistical Learning](#) », Springer, 2009.
- C.J. Huberty, S. Olejnik, « Applied MANOVA and Discriminant Analysis », Wiley, 2006.
- J.P. Nakache, J. Confais, « Statistique explicative appliquée », Technip, 2003.
- G. Saporta, « Probabilités, Analyse des Données et Statistique », Technip, 2006.
- R. Tomassone, M. Danzart, J.J. Daudin, J.P. Masson, « Discrimination et Classement », Masson, 1988.