COMP 6838 Data Mining

LECTURE 10:
Supervised classification

Dr. Edgar Acuna
Departmento de Matematicas

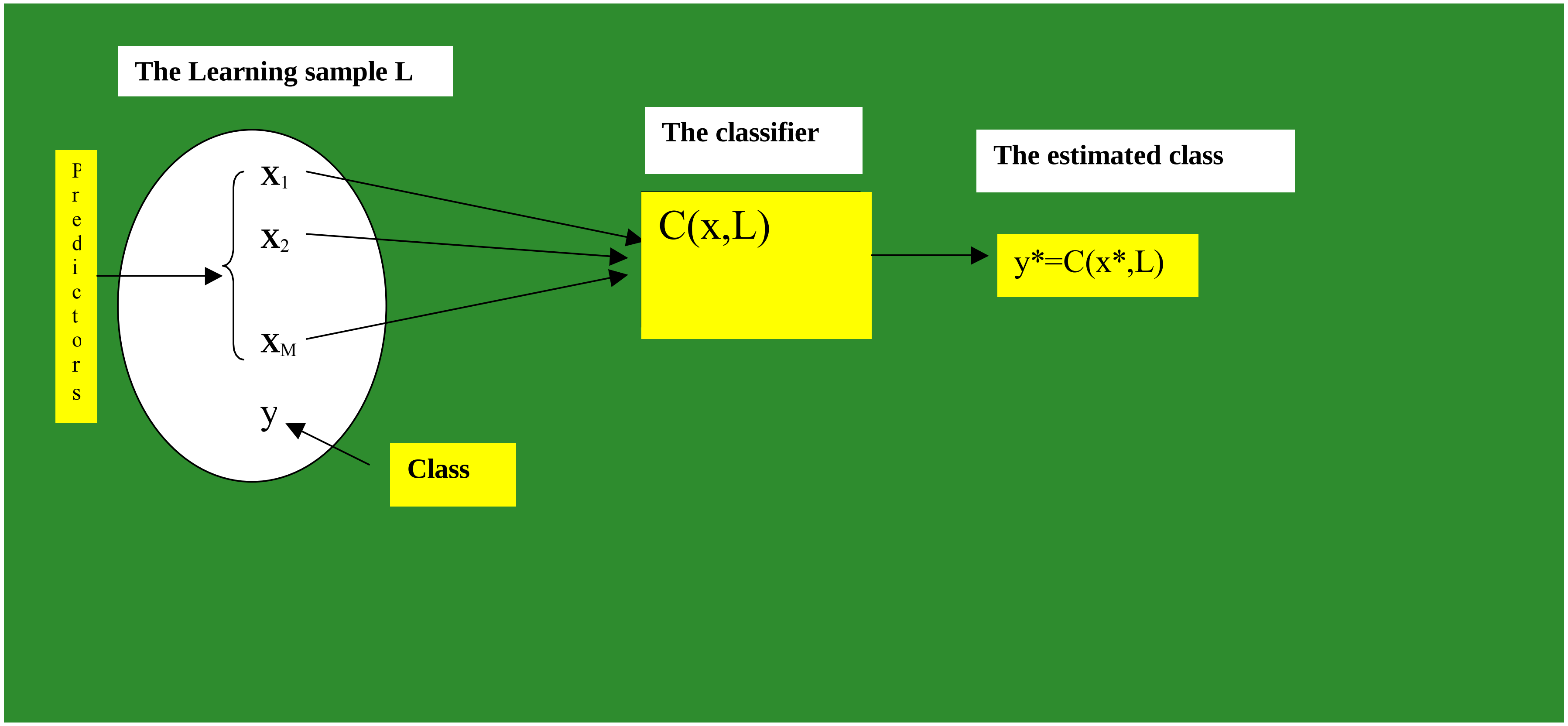
Universidad de Puerto Rico- Mayaguez

math.uprm.edu/~edgar

Supervised Classification vs. Prediction

- Supervised Classification:
  - predicts categorical class labels
  - classifies data (constructs a model) based on the training set and uses it in classifying new data
- Prediction (Regression):
  - models continuous-valued functions, i.e., predicts unknown or missing values
- Typical Applications
  - credit approval
  - target marketing
  - medical diagnosis
  - treatment effectiveness analysis

The Supervised classification problem



Supervised Classification—A Two-Step Process

- Model construction: describing a set of predetermined classes
  - Each tuple/sample is assumed to belong to a predefined class, as determined by the class label attribute
  - The model is represented as classification rules, decision trees, or mathematical formulae
- Model usage: for classifying future or unknown objects
  - Estimate accuracy of the model
    - The known label of test sample is compared with the classified result from the model
    - Accuracy rate is the percentage of test set samples that are correctly classified by the model
    - Test set is independent of training set, otherwise over-fitting will occur

Supervised classification methods

1. Linear Discriminant Analysis.
2. Nonlinear Methods: Quadratic Discrimination, Logistic Regression, Projection Pursuit.
3. Naive Bayes.
4. Decision Trees.
5. k-nearest neighbors
6. Classifiers based on kernel density estimation and gaussian mixtures.
7. Neural Networks: Multilayer perceptron. Radial Basis Function, Kohonen self-organizing map, Linear vector quantification.
8. Support vector machines.

Linear Discriminant Analysis

Consider the following training sample with p features and two classes

Y	**X_1**	**X_2**	**…**	**X_p**
1	X_{11}	X_{21}	….	X_{p1}
1	X_{12}	X_{22}	…	X_{p2}
..	..	..	..	..
1	X_{1n1}	X_{2n1}	…	X_{pn1}
2	$X_{1,n1+1}$	$X_{2,n1+1}$	…	$X_{p,n1+1}$
2	$X_{1,n1+2}$	$X_{2,n1+2}$	…	$X_{p,n1+2}$
..	..	..	…	..
2	$X_{1,n1+n2}$	$X_{2,n1+n2}$	…	$X_{p,n1+n2}$

Linear discriminant Analysis

Let $\overline{x}_1$ be the mean vector of the p features in class 1, and let $\overline{x}_2$ be the corresponding mean vector for the class 2.

Let us consider μ_1 and μ_2 as the mean vector of the respective class populations

Let us assume that both populations have the same covariance matrix, ie $\Sigma_1=\Sigma_2=\Sigma$. This is known as the homocedasticity property.

For now, we do not need to assume that the random vector of predictor **x**=(x1,…..xp) is normally distributed.

Linear discrimination is based on the following fact: An instance (object) x is assigned to the class C1 if

$$D(\mathbf{x}, C1)<D(\mathbf{x},C2) \quad (2.1)$$

where D(**x**,Ci)=, for i=1,2, represents the ***squared Mahalanobis distance*** *of* ***x*** *to the center of the Ci* ***class.***

Linear Discriminant Analysis (cont)

The expression (2.1) can be written as

$$(\mu_1 - \mu_2)'\Sigma^{-1}[x - (1/2)(\mu_1 + \mu_2)] > 0 \qquad (2.2)$$

Using the training sample, μ_i can be estimated by $\overline{x}_i$ and Σ is estimated by S, the pooled covariance matrix, which is calculated by

$$S = \frac{(n_1 - 1)S_1 + (n_2 - 1)S_2}{n_1 + n_2 - 2}$$

where, S1 and S2 represent the sample covariance matrices of the random vector of predictors in each class. Therefore the sample version of (2.2) is given by

$$(\overline{X}_1 - \overline{X}_2)'S^{-1}[x - (1/2)(\overline{X}_1 + \overline{X}_2)] > 0 \qquad (2.3)$$

The left hand side of expression (2.3) is called **the linear discriminant function.**

Example: Bupa(features 4 and 5)

```
> sigma1=cov(bupa[bupa[,7]==1,c(4,5)])
> sigma1
          V4        V5
V4  59.87759  143.1381
V5 143.13812 1103.9025
> sigma2=cov(bupa[bupa[,7]==2,c(4,5)])
> sigma2
         V4        V5
V4 127.4371  241.0319
V5 241.0319 1807.8202
> sigma=(144*sigma1+199*sigma2)/343
> sigma
          V4        V5
V4  99.07391  199.9336
V5 199.93361 1512.2979
>invsigma=solve(sigma)
```

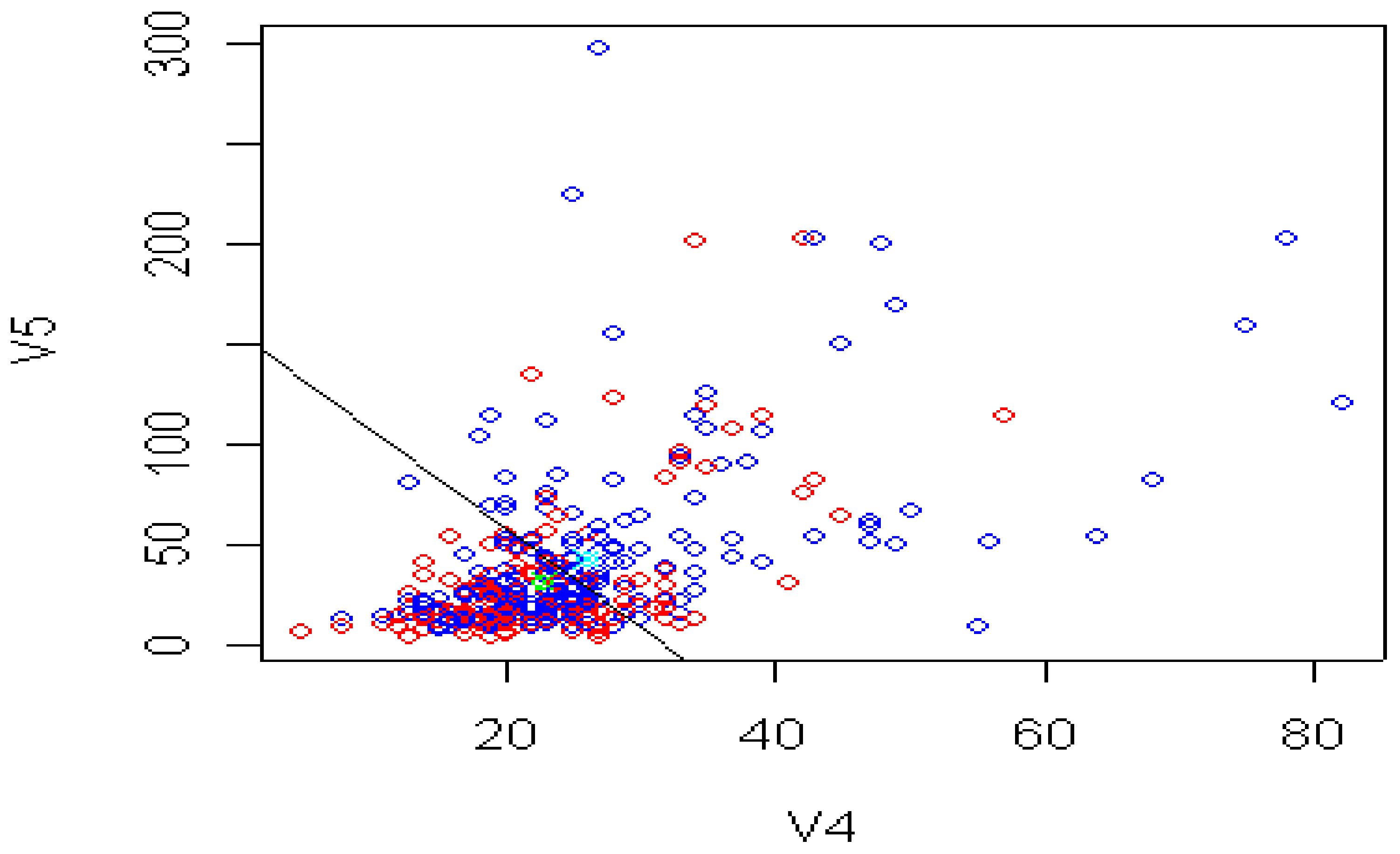
Example Bupa(features 4 and 5)

```
>invsigma
             V4           V5
V4  0.013766206 -0.001819964
V5 -0.001819964  0.000901854
> xbar1=mean(bupa[bupa[,7]==1,c(4,5)])
> xbar1
      V4       V5
22.78621 31.54483
> xbar2=mean(bupa[bupa[,7]==2,c(4,5)])
> xbar2
   V4    V5
25.99 43.17
> coeflda=t(xbar1-xbar2)%*%invsigma
> coeflda
              V4           V5
  [1,] -0.02294668 -0.004653421
```

Example Bupa(features 4 and 5)

```
>#Computing the independent term
>indlda=0.5*(xbar1-xbar2)%*%invsigma%*%(xbar1+xbar2)
>indlda
[1,] -0.7334659
>#Plot of the data points and the centroids of each class
>plot(bupa[bupa[,7]==1,4],bupa[bupa[,7]==1,5],col= "blue")
>points(bupa[bupa[,7]==2,4],bupa[bupa[,7]==2,5],col= "red")
>points(xbar1[1],xbar1[2],pch=19,col= "green")
>points(xbar2[1],xbar2[2],pch=19,col= "cyan")
>xbar1=mean(bupa[bupa[,7]==1,c(4,5)])
>#Plot of the discriminant linear function
> abline(indlda/coeflda[2],-coeflda[1]/coeflda[2])
```

LDA for Bupa based on features 4 and 5



LDA(Fisher, 1936)

Fisher obtained the linear discriminant function of equation (2.3) but following other way. He tried to find a linear combination of the features x's that separates classes C1 and C2 at most as possible under the assumption of homogeneity of covariance matrices (Σ1=Σ2=Σ). More specifically, if y=**d'x** then, the squared distance between the mean of y in each class divided by the varianza of y in each group is given by

$$\frac{(d'\mu_1 - d'\mu_2)^2}{d'\Sigma d} \qquad (2.4).$$

This ratio is maximized when d=$\Sigma^{-1}(\mu_1 - \mu_2)$. This result is obtained by an application pf the Cauchy-Schwartz's inequality (See Rao, C. R. *Linear Statistical Inference and its applications*, p. 60).

LDA (cont)

- The numerator is also called the sum of squares between groups (BSS), and the denominator is called the sum of squares within groups (WSS). An estimate of the **d** value is $S^{-1}(\bar{x}_1 - \bar{x}_2)$.
- Fisher asigned an object **x to** class C1 if y=$(\bar{x}_1 - \bar{x}_2)'$ S^{-1} **x** is closer to $\bar{y}_1 = (\bar{x}_1 - \bar{x}_2)' S^{-1} \bar{x}_1$ than to $\bar{y}_2$. The midpoint between $\bar{y}_1$ y $\bar{y}_2$ is
$$\frac{(\bar{x}_1 - \bar{x}_2)' S^{-1} (\bar{x}_1 - \bar{x}_2)}{2}$$
- Notice that y is closer $\bar{y}_1$ if y> $\frac{\bar{y}_1 + \bar{y}_2}{2}$, this yields the equation (2.3).

Tests for homogeneity of covariance matrices(homocedasticity)

- The most well known test for cheking homocedasticity (homogeneity of covariance matrices) is the Bartlett test. This test is a modification of the likelihood ratio test, however it is subject to the assumption of multivariate normal distribution. It makes use of a Chi-Square distribution. Bartlett test is available in SAS. The Mardia test is one of several test to check multivariate normality.
- Other alternative is to extent the Levene’s test for comparing the variance of several univariate populations.
- Some statistical packages like SPSS use the Box’M test to check homocedasticity).

The Van Valen's test for homocedasticity

- It is easy to implement and requires only the use of a two-sample t-test. First, each feature need to be standarized and then the following values are computed

$$d_{ij} = \sqrt{\sum_{k=1}^{p} (x_{ijk} - M_{jk})^2}$$

where x_{ijk} is the value of i-th instance for the k-th feature in the j-th group. M_{jk} is the median of the j-th feature in the j-th sample. Finally the sample mean of the dij's for the j-th groups are compared. For datasets with two classes, a two sample t-test assuming unequal variance can be used. For more than two classes a F-test is needed. However is better to use the corresponding nonparametric tests: Wilcoxon and Kruskal-Wallis. The null hypothesis is that there is not homocedasticity.

```
> vvalen(my.iris)

        Kruskal-Wallis rank sum test

data:  testlist
Kruskal-Wallis chi-squared = 2.1107, df = 2, p-value = 0.3481
```

Se acepta la hipotesis Nula No hay homocedasticidad.

```
> vvalen(bupa)

        Kruskal-Wallis rank sum test

data:  testlist
Kruskal-Wallis chi-squared = 10.1621, df = 1, p-value = 0.001434
```

Se rechaza la hipotesi Nula. Si hay Homocedasticicidad.

Mardia’s test for multivariate Normality (1970)

Consideremos que **x’j** (j=1,….n) representan las observaciones en la muestra de entrenamiento correspondiente a una clase particular C. Si se consideran p variables predictoras entonces cada **xj** es un vector columna p-dimensional. Deseamos probar que el vector aleatorio **X**=(X1,…..Xp) se distribuye en forma normal multivariada en C. Mardia basa su prueba en las medidas de asimetría y kurtosis, cuyas estimaciones basadas en la muestra de entrenamiento están definidas como :

$$b_1 = \frac{1}{n^2} \sum_{j=1}^{n} \sum_{k=1}^{n} \{ (\mathbf{x}_j - \bar{\mathbf{x}})' \mathbf{S}^{-1} (\mathbf{x}_k - \bar{\mathbf{x}}) \}^3$$

y

$$b_2 = \frac{1}{n} \sum_{j=1}^{n} \{ (\mathbf{x}_j - \bar{\mathbf{x}})' \mathbf{S}^{-1} (\mathbf{x}_j - \bar{\mathbf{x}}) \}^2$$

respectivamente.

Mardia’s test (cont)

Si la hipótesis nula Ho: **x** es normal multivariada en la clase C es cierta entonces se puede mostrar que para n grande

$$(n/6)b_1 \sim \chi^2_d$$

con *d*=(p/6)(p+1)(p+2) grados de libertad, y

$$[b_2 - p(p+2)]/\sqrt{(8/n)p(p+2)} \sim N(0,1)$$

- La prueba de Hawkins (Technometrics, 1981) permite probar simultáneamente normalidad multivariada y homocedasticidad. No aparece en ningún programa estadístico.

Example: Iris

```
> mardia(miris,1)
mard1= 24.15508
pvalue for m3= 0.2356838
mard2= 0.7587116
p-value for m4= 0.4480251
There is  statistical evidence for normality
> mardia(miris,2)
mard1= 23.70393
pvalue for m3= 0.2555643
mard2= -1.034219
p-value for m4= 0.3010336
There is statistical evidence for normality
> mardia(miris,3)
mard1= 24.72568
pvalue for m3= 0.2121282
mard2= -0.3384283
p-value for m4= 0.7350404
There is  statistical evidence for normality
```

Example:Bupa

```
> mardia(bupa,1)
mard1= 420.9489
pvalue for m3= 0
mard2= 15.91613
p-value for m4= 0
There is not statistical evidence for normality
> mardia(bupa,2)
mard1= 1178.14
pvalue for m3= 0
mard2= 37.50413
p-value for m4= 0
There is not statistical evidence for normality
```

Supervised classification from a Bayesian point of view

Suposse that we known beforehand the prior probabilities πi (i=1, 2,…G) that an object belongs to the class Ci . If n o any additional information then the best decision rule will classify the objetc as belonging to the class Ci if

$$\pi i > \pi j \quad \text{for} \quad i=1,2,\ldots.G, \; i \neq j \qquad (3.1)$$

However, usually some addtional information is known, such as a vector of measurements x made on the object to be classified. In this case we compare the probabality of belongonng to each class for an object with vecotr of measurements x and the object is classified as of class Ci if

$$P(Ci/\mathbf{x})>P(Cj/\mathbf{x}) \text{ para todo } i \neq j \qquad (3.2)$$

This decision rule is called the **Bayes rule of minimum error**.

Notice that i=argmax$_k$ P(Ck/**x**) for all k in 1,2….G.

Bayesian Classification

The probabillities P(Ci/x) are called posterior probabilities. Unfortunately rarely the posterior probabiities are known and they must be estimated. This ocuurs in logistic regression, decision trees classifiers, and neural networks.

A more convenient formulation of the former rule can be obtained by applying Bayes Theorem, which states that

$$P(Ci/\mathbf{x}) = \frac{f(\mathbf{x}/C_i)\pi_i}{f(\mathbf{x})} \qquad (3.3)$$

Therefore an object will be classified as of class Ci if

$$f(\mathbf{x}/C_i)\pi_i > f(\mathbf{x}/C_j)\pi_j \qquad (3.4)$$

para todo i $\neq$j. That is, i=argmax$_k$f(x/C$_k$)π_k

If the class conditional densities *f(**x**/ Ci) are known then the classification problem is solved, like it occurs in both linear and quadratic discriminant.*
But sometimes the *f(**x**/ Ci) are unknown and they must be estimated sing the training sample. This is the case of k-nn classifiers, kernel density classifiers and gaussian mixtures classifiers.*

Linear discriminant analysis as a Bayesian classifier

Let us consider that we have two classes C1 y C2 that follow a multivariate normal distribution, Np($\mathbf{u}_1,\Sigma_1$) and Np($\mathbf{u}_2,\Sigma_2$) respectively y que además tienen igual matriz de covarianza $\Sigma_1=\Sigma_2=\Sigma$. Then the equation 3.4 can be written as

$$\frac{\exp[-1/2(\mathbf{x}-\mathbf{u}_1)'\Sigma^{-1}(\mathbf{x}-\mathbf{u}_1)}{\exp[-1/2(\mathbf{x}-\mathbf{u}_2)'\Sigma^{-1}(\mathbf{x}-\mathbf{u}_2)}>\frac{\pi_2}{\pi_1} \qquad (3.5)$$

Taking logarithms in both sides, one obtains

$$-1/2[(\mathbf{x}-\mathbf{u}_1)'\Sigma^{-1}(\mathbf{x}-\mathbf{u}_1)-(\mathbf{x}-\mathbf{u}_2)'\Sigma^{-1}(x-\mathbf{u}_2)]>Ln(\frac{\pi_2}{\pi_1}) \qquad (3.6)$$

After some simplifications one gets

$$(u_1-u_2)'\Sigma^{-1}(x-1/2(u_1+u_2)) > Ln\left(\frac{\pi_2}{\pi_1}\right) \quad (3.7)$$

This inequality is similiar to the one given in (2.2), except by the term in the right hand side, but if we estimate the population parameters and in addition we consider that the prior probabilities are equal ($\pi_1=\pi_2$) then both expressions become the same.

LDA for more than two classes

For G classes, the LDA assigns an object with attributes vector x to the class i such that

$$i=\text{argmax}_k\ \mu'_k\Sigma^{-1}x-(1/2)\ \mu'_k\Sigma^{-1}\ \mu'_k\ +\text{Ln}(\pi_k)$$

For all k in 1,2,…G. As before the right hand-side is estimated using the training sample.

Example: Vehicle dataset

```
Library(MASS)
ldaveh=lda(vehicle[,1:18],vehicle[,19])
predict(ldaveh)$posterior
predict(ldaveh)$class
# Estimating the error rate
mean(vehicle[,19]!=predict(ldaveh)$class)
[1] 0.2021277
```

It is estimated that 20.21% of instances misclassified.

Quadratic discriminant analysis

It does not require the homocedasticity condition.

```
 qdaveh=qda(vehicle[,1:18],vehicle[,19])
> mean(vehicle[,19]!=predict(qdaveh)$class)
[1] 0.08392435
>
```

Notice that for the vehicle dataset qda performs better than lda

The misclassification error rate

The misclassification error rate R(d) is the probability that the classifier d classifies incorrectly an instance coming from a sample (test sample) obtained in a later stage than the training sample. Also is called the True error or the actual error.
It is an unknown value that needs to be estimated.

Methods for estimation of the misclassification error rate

i) **Resubstitution or Aparent Aparente** (Smith, 1947). This is merely the proportion of instances in the training sample that are incorrectly classified by the classification rule. In general is an estimator too optimistic and it can lead to wrong conclusions if the number of instances is not large compared with the number of features. This estimator has a large bias.

ii) **"Leave one out" estimation. (**Lachenbruch, 1965). In this case an instance is omitted from the training sample. Then the classifier is built and the prediction for the omitted instances is obtained. One must register if the instance was correctly or incorrectly classfied. The process is repeated for all the instances in the training sample and the estimation of the ME will be given by the proportion of instances incorrectly classified. This estimator has low bias but its variance tends to be large.

Examples of LOO

```
ldaveh=lda(vehicle[,1:18],vehicle[,19],CV=TRUE)
> mean(vehicle[,19]!=ldaveh$class)
[1] 0.2210402
>
> ldabupa=lda(bupa[,1:6],bupa[,7])
> mean(bupa[,7]!=predict(ldabupa)$class)
[1] 0.2956522
> ldabupa1=lda(bupa[,1:6],bupa[,7],CV=TRUE)
> mean(bupa[,7]!=ldabupa1$class)
[1] 0.3014493
>
```

Methods for estimation of the misclassification error rate

iii) Cross validation. (Stone, 1974) In this case the training sample is randomly divided in v parts (v=10 is the most used). Then the classifier is built using all the parts but one. The omitted part is considered as the test sample and the predictions for each instance on it are found. The CV misclassification error rate is found by adding the misclassification on each part and dividing them by the total number of instances. The CV estimated has low bias but high variance. In order to reduce the variability we usually repeat the estimation several times.

The estimation of the variance of the CV estimator is a hard problem (Bengio and Grandvalet, 2004).

Example:

```
 cv10lda(vehicle,repet=10)
[1] 0.2192671
> cv10lda(vehicle,repet=20)
[1] 0.2206856
>
```

iv) The holdout method. A percentage (70%) of the dataset is considered as the training sample and the remaining as the test sample. The classifier is evaluated in the test sample. The experiment is repeated several times and then the average is taken.

v) Bootstrapping. (Efron, 1983). In this method we generate several training samples by sampling with replacement from the original training sample. The idea is to reduce the bias of the resubstitution error.

It is almost unbiased, but it has a large variance. Its computation cost is high. There exist several variants of this method.