LABORATORIO DE ANÁLISIS DISCRIMINANTE EN MATLAB

Este ejercicio consiste en aplicar el método análisis discriminante como herramienta para clasificar variables independientes mediante funciones de MATLAB. En este caso la prueba será con dos variables previamente clasificadas y se corresponde con el incluido en las notas disponibles en este sitio.

La tabla financiera de veinticuatro empresas, de las cuales hay doce aventajadas y las otra doce menos, muestra dos variables x donde x_1 se corresponde con la razón financiera *ganancias antes de los intereses a los activos totales* y x_2 con la razón financiera *el retorno al capital total*.

Grupo 1: Más aventajadas				Grupo2: Menos aventajadas			
Empresa	x1	x2	Z	Empresa	x1	x2	Z
1	0,158	0,182	0,240	13	-0,012	-0,031	-0,030
2	0,210	0,206	0,294	14	0,036	0,053	0,063
3	0,207	0,188	0,279	15	0,038	0,036	0,052
4	0,280	0,236	0,365	16	-0,063	-0,074	-0,097
5	0,197	0,193	0,276	17	-0,054	-0,119	-0,122
6	0,227	0,173	0,283	18	0,000	-0,005	-0,004
7	0,148	0,196	0,243	19	0,005	0,039	0,031
8	0,254	0,212	0,329	20	0,091	0,122	0,151
9	0,079	0,147	0,160	21	-0,036	-0,072	-0,076
10	0,149	0,128	0,196	22	0,045	0,064	0,077
11	0,200	0,150	0,247	23	-0,026	-0,024	-0,035
12	0,187	0,191	0,267	24	0,016	0,026	0,030

El ejercicio consiste en crear un procedimiento en MATLAB que permita clasificar satisfactoriamente a un grupo de observaciones nuevas de acuerdo a una función discriminante.

El procedimiento emplea la función:

[clase,error] =classify(prueba,entrenamiento,grupo observado)

donde
clase: es el resultado de clasificar los valores nuevos dispuestos en la matriz prueba
error: nivel de error en clasificación

Se carga el archivo de datos que se corresponde con la tabla anterior

load 'Rfempresas.txt'

datos=Rfempresas;
ans =

12 8

Organizamos las observaciones con la primera razón financiera en X

X=[datos(:,2);datos(:,6)];

y la segunda razón financiera en Y por

Y=[datos(:,3);datos(:,7)];

Definimos dos grupos: Más aventajados (2) y menos aventajados (1). Los primeros doce elementos de cada variable X y Y son del grupo 2 y los otros doce, son del grupo 1

Definimos el grupo 2 (menos aventajados) con uno

grupo2=ones(12,1);

y el grupo 2 con los más aventajados (2)

grupo1=[2 2 2 2 2 2 2 2 2 2 2 2]';

size(grupo1)
ans =

12 1

size(grupo2)
ans =

12 1

Extraemos dos arreglos correspondientes a las observaciones de prueba y de entrenamiento a partir del arreglo original **datos**.

Para las variables X (más aventajados) y Y (menos aventajados)

X'
ans =

Columns 1 through 12

0.1580 0.2100 0.2070 0.2800 0.1970 0.2270 0.1480 0.2540 0.0790 0.1490 0.2000 0.1870

Columns 13 through 24

-0.0120 0.0360 0.0380 -0.0630 -0.0540 0 0.0050 0.0910 -0.0360 0.0450 -0.0260 0.0160

Y'
ans =

Columns 1 through 12

0.1820 0.2060 0.1880 0.2360 0.1930 0.1730 0.1960 0.2120 0.1470 0.1280 0.1500 0.1910

Columns 13 through 24

-0.0310 0.0530 0.0360 -0.0740 -0.1190 -0.0050 0.0390 0.1220 -0.0720 0.0640 -0.0240 0.0260

Seleccionamos catorce para la matriz de entrenamiento (E) y diz para la matriz de prueba (P)

size(X)
ans =

24 1

size(Y)
ans =

24 1

Dejamos definidos **E** (datos de entrenamiento), **P** (datos de prueba) y faltaría definir el vector grupo que se corresponde con los valores observados a las clases (2 o 1). Este vector debe estar asociado a la matriz de entrenamiento y debe tener a misma dimensión

E=[[X(1:6);X(13:20)] [Y(1:6);Y(13:20)]];

P=[[X(7:12);X(21:24)] [Y(7:12);Y(21:24)]];

size(E)
ans =

14 2

size(P)
ans =

10 2

Para la definición de los grupos, seleccionamos inicialmente dos vectores, el primero que contiene la clase 1 (menos aventajados)

El vector de grupo sería la agrupación de ambos grupo1 y grupo2

grupo1=ones(8,1);

grupo2= ones(6,1)+ones(6,1);

grupo=[grupo2;grupo1];

size(grupo)
ans =

14 1

Ya organizados todos los datos originales, tenemos los siguientes arreglos

E

0.1580 0.1820
0.2100 0.2060
0.2070 0.1880
0.2800 0.2360
0.1970 0.1930
0.2270 0.1730
-0.0120 -0.0310

```
    0.0360    0.0530
    0.0380    0.0360
   -0.0630   -0.0740
   -0.0540   -0.1190
         0   -0.0050
    0.0050    0.0390
    0.0910    0.1220
```

P

```
    0.1480    0.1960
    0.2540    0.2120
    0.0790    0.1470
    0.1490    0.1280
    0.2000    0.1500
    0.1870    0.1910
   -0.0360   -0.0720
    0.0450    0.0640
   -0.0260   -0.0240
    0.0160    0.0260
```

grupo'

```
     2     2     2     2     2     2     1     1     1     1     1     1     1     1
```

Para comparar los resultados definamos a **grupop** como el arreglo que contiene los valores reales de clasificación de las nuevas observaciones usadas para hacer la prueba.

grupop=[ones(6,1)+ones(6,1);ones(4,1)];

grupop'

```
     2     2     2     2     2     2     1     1     1     1
```

Empleamos la función de análisis discriminante lineal

>> help classify

CLASSIFY Discriminant analysis.

CLASS = CLASSIFY(SAMPLE,TRAINING,GROUP) classifies each row of the data in SAMPLE into one of the groups in TRAINING. SAMPLE and TRAINING must be matrices with the same number of columns. GROUP is a grouping variable for TRAINING. Its unique values define groups, and each element defines which group the corresponding row of TRAINING belongs to. GROUP can be a numeric vector, a string array, or a cell array of strings. TRAINING and GROUP must have the same number of rows. CLASSIFY treats NaNs or empty strings in GROUP as missing values, and ignores the corresponding rows of TRAINING.

CLASS indicates which group each row of SAMPLE has been assigned to, and is of the same type as GROUP.

[CLASS,ERR] = CLASSIFY(SAMPLE,TRAINING,GROUP) also returns an estimate of the misclassification error rate. CLASSIFY returns the apparent error rate, i.e., the percentage of observations in the TRAINING that are misclassified.

CLASS = CLASSIFY(SAMPLE,TRAINING,GROUP,TYPE) allows you to specify the typeof discriminant function, one of 'linear', 'quadratic', or 'mahalanobis'.

Linear discrimination fits a multivariate normal density to each group, with a pooled estimate of covariance. Quadratic discrimination fits MVN densities with covariance estimates stratified by group. Mahalanobis discrimination uses Mahalanobis distances with stratified covariance estimates. TYPE defaults to 'linear'.

CLASS = CLASSIFY(SAMPLE,TRAINING,GROUP,TYPE,PRIOR) allows you to specify prior probabilities for the groups in one of three ways. PRIOR can be a numeric vector of the same length as the number of unique values in GROUP.

If GROUP is numeric, the order of PRIOR must correspond to the sorted values in GROUP, or, if GROUP contains strings, to the order of first

occurrence of the values in GROUP. PRIOR can also be a 1-by-1 structure with fields 'prob', a numeric vector, and 'group', of the same type as GROUP, and containing unique values indicating which groups the elements of 'prob' correspond to. As a structure, PRIOR may contain groups that do not appear in GROUP. This can be useful if TRAINING is a subset a larger training set. Finally, PRIOR can also be the string value 'empirical', indicating that the group prior probabilities should be estimated from the group relative frequencies in TRAINING. PRIOR defaults to a numeric vector of equal probabilities, i.e., a uniform distribution. PRIOR is not used for discrimination by Mahalanobis distance, except for error rate calculation.

Examples:

```
% training data: two normal components
   training = [mvnrnd([ 1  1],   eye(2), 100); ...
               mvnrnd([-1 -1], 2*eye(2), 100)];
   group = [repmat(1,100,1); repmat(2,100,1)];
% some random sample data
   sample = unifrnd(-5, 5, 100, 2);

% classify with a known prior, 30% prob of group 1, 70% of group 2
   c = classify(sample, training, group, 'quad', [.3 .7]);
```

[clase,error]=classify(P,E,grupo);

clase'

ans =

2 2 1 2 2 2 1 1 1 1

grupop'

ans =

2 2 2 2 2 2 1 1 1 1

De acuerdo a los resultados hubo un nivel de acierto de aproximadamente 90%. La tercera observación no fue bien clasificada.

Veamos algunas estadísticas y diagramas.

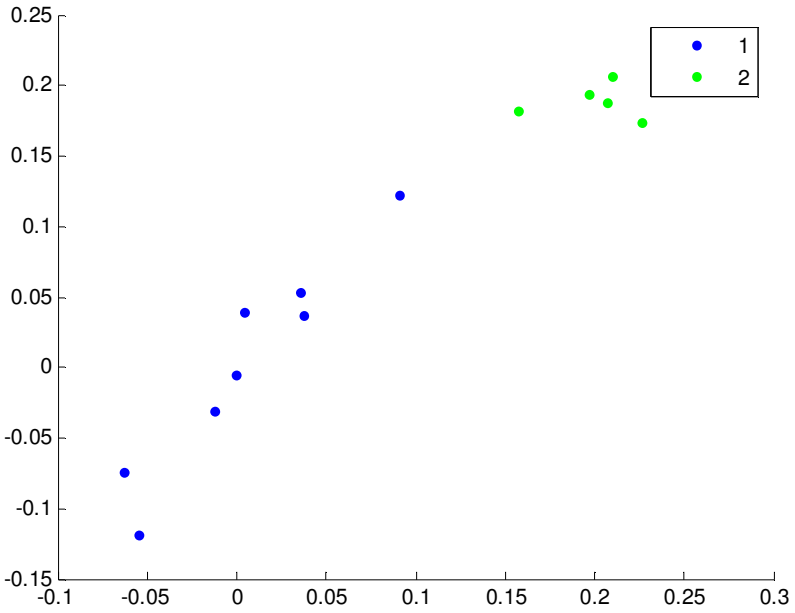
Observemos nuestras variables en esta sesión de MATLAB
who

Your variables are:

E Rfempresas X2 Y clases grupo grupon sample train2
P X XT ans datos grupo1 grupop train
RFempresas X1 XY clase error grupo2 grupos train1

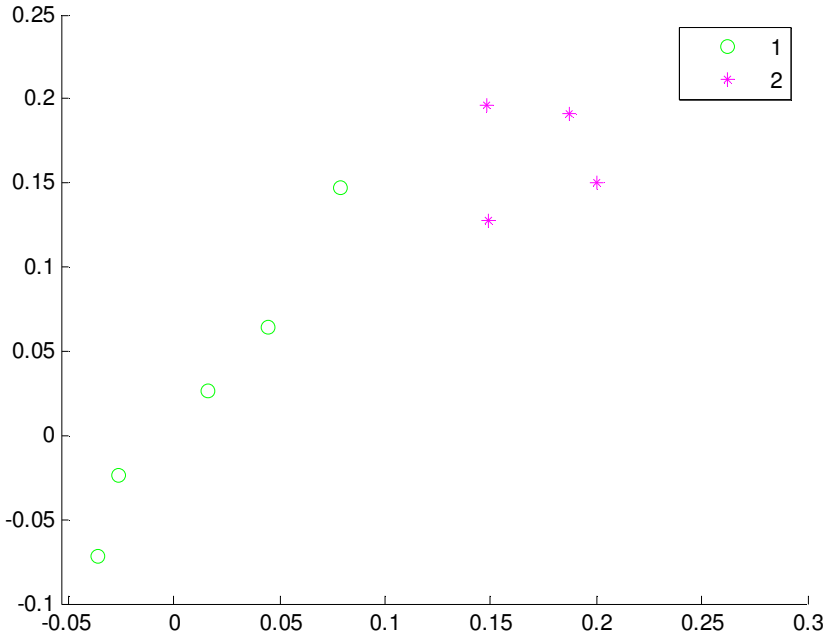
Las observaciones empleadas para la determinación de la función discriminante están representadas en el siguiente gráfico.

gscatter(E(:,1),E(:,2),grupo)



Para los grupos de prueba se seleccionaron algunas opciones de la función **gscatter** y se representaron los resultados de la función.

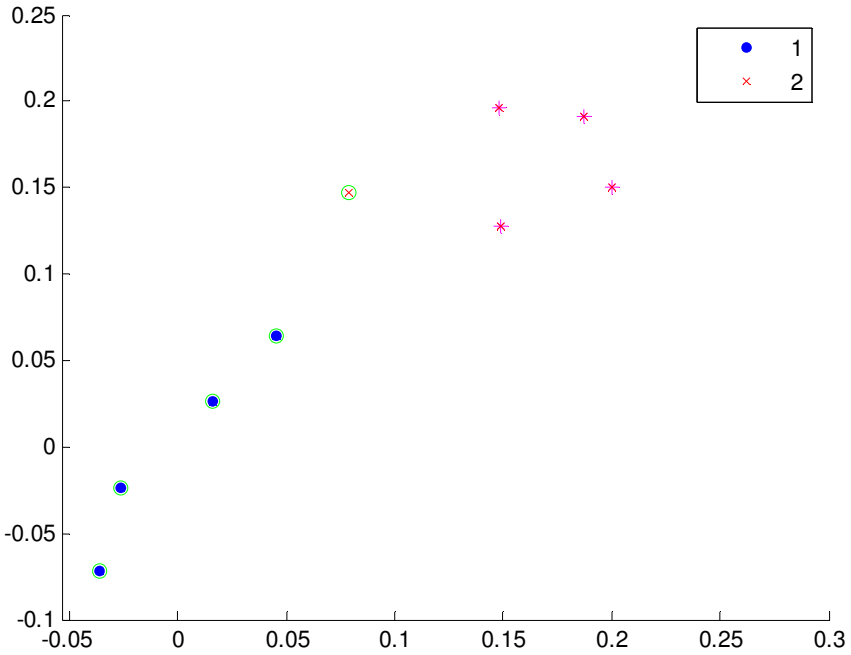
gscatter(P(:,1),P(:,2),clase,'gm','o*')
hold
Current plot held



Grupo de prueba con valores estimados

Al ser comparados con los valores realmente observados tenemos la siguiente superposición de gráficas

gscatter(P(:,1),P(:,2),grupop,'br','.x')



Veamos ahora que tan buena fue la clasificación. De acuerdo a lo estimado y a lo observado, el mismo conjunto de observaciones se agrupan en el arreglo balance, donde la primera columna son los valores observados de los grupos y en la segunda, los valores estimados

balance=[grupop(:) clase(:)];

Para determinar las frecuencias de cada combinación posible de los dos niveles de clasificación, se emplea la función **crosstab**. Esta función contabiliza las observaciones para cada combinación de clases: 11,12, 21, 22.

help crosstab

CROSSTAB Cross-tabulation of column vectors.
CROSSTAB(COL1,COL2,...) takes vectors, string arrays, or cell arrays of strings, and returns an array, TABLE, of crosstabs. Element (i,j,...) of TABLE contains the count of all instances where COL1 = i, COL2 = j, and so on.
[TABLE, CHI2, P] = CROSSTAB(...) also returns the chisquare statistic, CHI2, for testing independence of each dimension of TABLE. (That is, that the proportion of items falling in any cell is equal to the product of the

proportion in that row, times the proportion in that column, and so on.) The scalar, P, is the significance level of the test. Values of P near zero cast doubt on the assumption of independence.
[TABLE, CHI2, P, LABELS] = CROSSTAB(...) also returns a cell array of labels for TABLE. The entries in the first column of LABELS are labels for the rows of TABLE, the entries in the second column are labels for the columns, and so on.

Reference page in Help browser doc crosstab

La frecuencia por cada columna es como sigue.

tabulate(balance(:,1))

Value	Count	Percent
1	4	40.00%
2	6	60.00%

>> tabulate(balance(:,2))

Value	Count	Percent
1	5	50.00%
2	5	50.00%

El arreglo con los valores a comparar:

>> balance

balance =

2	2
2	2
2	1
2	2
2	2
2	2

```
    1     1
    1     1
    1     1
    1     1
```

La función para estimar las frecuencias individuales

>> confusion=crosstab(balance(:,1),balance(:,2))

Determinamos los aciertos y errores en la clasificación en la matriz confusión.

```
confusion =
     4     0
     1     5
```

La frecuencia de cada fila determinará la razón y porsteriormente el porcentaje de confusión para cada caso.

totales= [sum(confusión(:,1:2))];

porcentaje=[confusion(1,:)./totales(1);confusion(2,:)./totales(2)]

```
porcentaje =

    1.0000         0
    0.1667    0.8333
```

En porcentajes sería

porcentaje=[confusion(1,:)./totales(1);confusion(2,:)./totales(2)]*100

```
porcentaje =

  100.0000         0
   16.6667   83.3333
```

Este porcentaje nos indica que hubo un 100% de acierto de los que corresponde con las empresas menos aventajadas. Por el contrario, no fue así con las más aventajadas. Sólo un 83.33% fue acertado en la clasificación.

>>