UNIDAD 1. INTRODUCCIÓN A LA MINERÍA DE DATOS. ANALISIS CLUSTER. K-Means. EM (Expectation- Maximization)

INTRODUCCIÓN A LA MINERÍA DE DATOS

Siguiendo a Maimon (2010) el descubrimiento de conocimiento en bases de datos (KDD, Knowledge Discovery in Databases) consiste en el análisis automático exploratorio y modelado de grandes conjuntos de datos. KDD es el proceso organizado de identificación de patrones válidos, novedosos, útiles, y comprensibles, a partir de conjuntos de datos grandes y complejos.

La minería de datos (Data Mining) es el núcleo del proceso KDD, y consiste en el desarrollo de algoritmos que exploran los datos, desarrollan el modelo y descubren patrones previamente desconocidos. El modelo se puede utilizar para entender los fenómenos a partir de los datos, para su análisis, y para la predicción o identificación.

El proceso KDD incluye en general nueve pasos:

1. Comprender el dominio de la aplicación y establecer los objetivos.
2. Seleccionar o construir el conjunto de datos.
3. Preprocesamiento y depuración de datos (manejo de valores perdidos, eliminación de ruidos o valores atípicos…).
4. Transformación de datos (reducción de dimensión, transformación de atributos, discretización de variables numéricas, transformación funcional, …).
5. Elegir la tarea apropiada de minería de datos (clasificación, regresión, identificación,…).
6. Elección del algoritmo de minería de datos.
7. Aplicación del algoritmo de minería de datos.
8. Evaluación.
9. Utilización del conocimiento descubierto.

Hay muchos métodos de minería de datos utilizados para diferentes propósitos y objetivos. Es útil distinguir entre dos tipos principales de minería de datos: orientado a la verificación (el sistema verifica las hipótesis del usuario) y orientado al descubrimiento (el sistema encuentra nuevas reglas y patrones de forma automática).

A su vez los métodos orientados al descubrimiento se subdividen en métodos de predicción y métodos de descripción. Los métodos descriptivos tienen como objetivo la interpretación de datos, que se enfoca hacia la comprensión, generalmente por visualización, mediante representación gráfica. Los métodos orientados a la predicción intentan construir automáticamente un patrón o modelo de comportamiento capaz de predecir valores.

La mayoría de las técnicas de minería de datos están orientadas al descubrimiento, y se basan en el aprendizaje inductivo, mediante el cual se construye un modelo explícito o implícito generalizando a partir de un número suficiente de casos de entrenamiento.

Los métodos de verificación, por otro lado, intentan la evaluación de una hipótesis propuesta generalmente por una fuente externa (como un experto). Estos métodos se relacionan con los más comunes de la estadística tradicional, como pruebas de bondad

de ajuste, contraste de hipótesis, y análisis de varianza, y se asocian menos con la minería de datos que sus homólogos orientados al descubrimiento, debido a que la mayoría de los problemas de Data Mining están relacionados con la búsqueda de una hipótesis en lugar de probar una conocida. Gran parte del enfoque de los métodos estadísticos tradicionales se basa en la estimación de un modelo predefinido, en oposición a uno de los principales Objetivos de Data Mining: identificación del modelo y su construcción basada en la evidencia, sin que se haya definido un modelo previo.

La terminología habitual utilizada por la comunidad de aprendizaje automático describe a los métodos de predicción como aprendizaje supervisado, a diferencia del aprendizaje no supervisado correspondiente a los métodos de descripción. El aprendizaje no supervisado se refiere a la modelización de la distribución de casos en un típico espacio de entrada de alta dimensión, e incluye principalmente técnicas que agrupan casos sin un atributo dependiente preespecificado. Los métodos supervisados, por otra parte, intentan descubrir la relación entre los atributos de entrada (variables independientes o explicativas) y un atributo de destino (variable dependiente), relación que se representa mediante una estructura que denominamos modelo. Usualmente los modelos describen y explican fenómenos que están ocultos en el conjunto de datos, y se pueden utilizar para predecir el valor del atributo de destino conociendo los valores de los atributos o variables de entrada.

Es útil distinguir entre dos modelos principales supervisados: modelos de clasificación y modelos de regresión; los primeros asignan el espacio de entrada a clases predefinidas, e intentan predecir la clase a la que pertenece cada elemento, mientras que los modelos de regresión asignan el espacio de entrada a un dominio de variable real, e intentan predecir el valor de esa variable.

Existen actualmente cientos de algoritmos de Minería de Datos ¿Cuál de ellos es mejor? La comparación empírica del rendimiento de los distintos métodos ha mostrado que cada uno funciona mejor en algunos entornos, y peor en otros. Se acepta que ningún algoritmo de inducción puede ser el mejor en todos los casos, debido a que cada uno de ellos tiene algún sesgo que lo lleva a preferir ciertas generalizaciones sobre otras, y tendrá éxito siempre que su sesgo coincida con las características del dominio de aplicación. El teorema NFL (No Free Lunch, no hay almuerzo gratis) presentado por David H. Wolpert y William G. Macready en 1995 demuestra que todos los algoritmos de optimización aplicados al conjunto de todos los problemas matemáticamente posibles, se comportan, en promedio, de la misma forma.

El error de generalización, el que se observa cuando el método se aplica al dominio general (a los casos nuevos no utilizados en el aprendizaje), es habitualmente mucho mayor que el obtenido cuando se aplica a los casos de la muestra, incluso con los mejores métodos, lo que revela un problema general de sobreajuste a los datos de la muestra. La investigación sobre este problema ha llevado a determinar el error mínimo alcanzable en el dominio de aplicación (conocido como error óptimo de Bayes); si los métodos existentes no alcanzan ese mínimo se requiere el desarrollo de nuevos algoritmos.

No existen en general normas válidas para preferir un método de minería de datos sobre otros, es decir una norma que funcione con todas las aplicaciones. En la práctica, para elegir el método más adecuado para utilizar en un problema determinado, es habitual definir una medida de rendimiento y aplicar diferentes métodos eligiendo finalmente el que parece funcionar mejor con los datos del estudio de acuerdo con esa medida.

En las seis unidades de este curso se describen los 12 métodos más frecuentes, considerados en general como los más útiles y eficaces en la mayoría de las aplicaciones reales; se trata de dos métodos de clasificación no supervisada K-MEANS, y EXPECTATION-MAXIMIZATION (EM), que se tratarán en esta primera unidad, ocho métodos de clasificación supervisada, algunos de los cuales pueden ser también utilizados como modelos de regresión: ADABOOST, RANDOM FOREST, CHAID, NAIVE BAYES (Clasificador Bayesiano), Arboles de Decisión (CART), Clasificación C5.0, KNN y SUPPORT VECTOR MACHINES (SVM), en las unidades 2 a 5, y finalmente, en la sexta y última unidad, el método A PRIORI utilizado para análisis de asociación, y un motor de búsqueda, PAGERANK, que puede ser también utilizado para la construcción de indicadores.

Los principios de funcionamiento son diferentes, aunque varios métodos comparten planteamientos y estrategias. Algunos permiten considerar únicamente variables explicativas cualitativas o variables cuantitativas, y otros admiten la combinación de variables explicativas de ambos tipos. Algunos utilizan una única variable dependiente de carácter cualitativo, que consiste generalmente en un factor de clasificación, mientras que otros permiten también la posibilidad de que la variable dependiente sea numérica.

K-MEANS. (K-medias)

K-medias (MacQueen, J. B., 1967) busca la mejor partición posible del conjunto de objetos para un número dado de clases k. Encuentra la clasificación que minimiza la suma de los cuadrados de las distancias de cada punto (cada elemento de la muestra) al centro de la clase a la que es asignado. Las variables deben ser numéricas.

El algoritmo comienza con el establecimiento de los centros iniciales de clase, que pueden ser k elementos de la muestra convenientemente separados entre sí, o simplemente k elementos elegidos al azar. Para cada clase tenemos por lo tanto un centro inicial. A continuación se asigna cada uno de los elementos de la muestra a la clase cuyo centro está más próximo, y finalizada esta asignación se recalculan los centros de cada clase como la media de todos los elementos que la forman. Estos dos pasos –asignación de elementos a clases y cálculo de los centros- se iteran hasta que la clasificación obtenida es estable. La regla de parada del algoritmo es múltiple: se termina cuando se repite la clasificación, cuando se ha alcanzado un número máximo de iteraciones establecido a priori, o bien cuando el cambio en el desplazamiento de los centros de clase entre dos iteraciones consecutivas es muy pequeño, menor que un valor umbral predeterminado.

Este algoritmo reduce en cada iteración la varianza interna de las clases, haciendo por lo tanto que las clases sean cada vez más homogéneas. Esta es una característica importante de la clasificación que se busca: los elementos de cada clase deben ser similares entre sí, formando clases homogéneas.

Con este método es necesario establecer a priori el número de clases que tendrá la solución buscada, aunque naturalmente se pueden obtener diferentes clasificaciones, con distinto número de grupos, aplicando el método de forma repetida.

El resultado puede depender en algunos casos de los centros iniciales elegidos, aunque en general la solución final es muy robusta frente a esta elección, y suele ser prácticamente independiente de los centros utilizados para iniciar el algoritmo. Algunos programas estadísticos utilizan diferentes arranques aleatorios y comprueban después si la solución obtenida es la misma con todos ellos.

Dado que el procedimiento consiste en reducir paulatinamente la variabilidad interna, el método obtiene clases con dispersión aproximadamente esférica. No es adecuado por lo tanto en aquellas aplicaciones en las que sabemos que las clases no tienen esa forma.

Es adecuado para clasificar conjuntos de elementos muy numerosos (por ejemplo con cientos de miles de objetos), con los cuales los métodos de clasificación jerárquica, que deben calcular en cada paso una matriz de distancias entre todos los elementos, son inviables dadas las dificultades de cálculo.

Se suele utilizar a continuación del método de clasificación del árbol mínimo (clasificación jerárquica), sirviendo éste para decidir el número de clases que se desea obtener. Si el conjunto de objetos es muy grande, es suficiente para este propósito –determinación del número idóneo de clases- una pequeña muestra aleatoria.

Entre los resultados de la aplicación del método de clasificación de k-medias, en general se obtienen los siguientes:

Número de elementos en cada clase.

Centros de clase: permiten identificar las características principales de las clases, interpretando los valores medios.

Varianza interna y total: Cuanto más pequeña sea la primera en relación a la segunda, mejor es la clasificación. También se puede mostrar la reducción de la varianza interna en cada paso o iteración del algoritmo.

Utilizaremos para su aplicación la función kmeans, del paquete básico stats (que no es necesario cargar). La sintaxis básica es muy simple:

kmeans(x, centers)

Debemos indicar el conjunto de datos x (data frame o matriz), y el número "centers" de clases que buscamos.

En el archivo algas.xls hay datos de la concentración relativa de 19 pigmentos en una muestra de 31 casos con 6 tipos de algas: Diatomeas, Clorofíceas, Dinofíceas, Eustigmatofíceas, Cianofíceas y Criptofíceas. El objetivo es determinar si es posible diferenciar las clases de algas en función de su composición relativa de pigmentos.

Para ello buscaremos la clasificación "natural" de nuestro conjunto con seis clases, utilizando para ello los 19 pigmentos, y después intentaremos relacionar esa clasificación con el tipo de alga.

Los datos utilizados en el curso deberán estar en la carpeta "C:/CURSO DM", descargados desde la página del curso. Desde R Commander configuramos en primer lugar el directorio de trabajo con Fichero → Cambiar directorio de trabajo, con el que elegimos esa carpeta "C:/CURSO DM". Alternativamente podemos copiar en la ventana de órdenes y ejecutar la orden siguiente, que hace exactamente lo mismo:

setwd("C:/CURSO DM")

A continuación leemos los datos desde el archivo con la opción del menú de R Commander Datos → Importar datos → desde un archivo de Excel. Cambiamos el nombre por defecto "Dataset" poniendo en su lugar "algas" y elegimos dentro de la carpeta de trabajo el archivo algas.xls. Obtendremos como resultado un conjunto de datos de R o data frame con el nombre "algas". { hay una copia de los datos en formato de R que se puede leer con load("algas.RData") }

Podemos ver los nombres de las 20 variables con la orden "names"

names(algas)

```
> names(algas)
 [1] "clase"     "chloroC2"  "chloroC1"  "methyl_c"  "peridin"   "fucoxant"
 [7] "neoxant"   "violaxan"  "diadinox"  "dinoxant"  "alloxant"  "zeaxanth"
[13] "lutein"    "chloropB"  "chl_b_epr" "chlAllome" "chlEpime"  "carotBu"
[19] "carotBe"   "carotBB"
```

Y un resumen descriptivo con la orden "summary":

summary(algas)

```
        clase            chloroC2         chloroC1           methyl_c
 cianofíceas     :4   Min.   :0.0000    Min.   :0.00000    Min.   :0.00000
 clorofíceas     :8   1st Qu.:0.0000    1st Qu.:0.00000    1st Qu.:0.00000
 criptofíceas    :4   Median :0.0000    Median :0.00000    Median :0.00000
 diatomeas       :6   Mean   :0.2968    Mean   :0.07323    Mean   :0.01226
 dinofíceas      :5   3rd Qu.:0.5850    3rd Qu.:0.00000    3rd Qu.:0.00500
 eustigmatoficeas:4   Max.   :1.1200    Max.   :0.85000    Max.   :0.11000

   peridin           fucoxant          neoxant           violaxan
 Min.   :0.0000     Min.   :0.0000    Min.   :0.00000    Min.   :0.00000
 1st Qu.:0.0000     1st Qu.:0.0000    1st Qu.:0.00000    1st Qu.:0.00000
 Median :0.0000     Median :0.0000    Median :0.00000    Median :0.00000
 Mean   :0.1929     Mean   :0.2777    Mean   :0.03677    Mean   :0.09903
 3rd Qu.:0.0000     3rd Qu.:0.0000    3rd Qu.:0.05500    3rd Qu.:0.06000
 Max.   :1.3900     Max.   :1.9900    Max.   :0.18000    Max.   :1.07000

   diadinox           dinoxant          alloxant         zeaxanth
 Min.   :0.00000    Min.   :0.00000   Min.   :0.0000    Min.   :0.00000
 1st Qu.:0.00000    1st Qu.:0.00000   1st Qu.:0.0000    1st Qu.:0.00000
 Median :0.00000    Median :0.00000   Median :0.0000    Median :0.00000
 Mean   :0.02226    Mean   :0.02452   Mean   :0.1216    Mean   :0.07516
 3rd Qu.:0.00000    3rd Qu.:0.01000   3rd Qu.:0.0000    3rd Qu.:0.11500
 Max.   :0.20000    Max.   :0.26000   Max.   :1.0700    Max.   :0.36000

    lutein            chloropB         chl_b_epr           chlAllome
 Min.   :0.0000    Min.   :0.00000    Min.   :0.0000000   Min.   :0.00000
 1st Qu.:0.0000    1st Qu.:0.00000    1st Qu.:0.0000000   1st Qu.:0.01000
 Median :0.0000    Median :0.00000    Median :0.0000000   Median :0.02000
 Mean   :0.1894    Mean   :0.05839    Mean   :0.0009677   Mean   :0.04548
 3rd Qu.:0.2700    3rd Qu.:0.09500    3rd Qu.:0.0000000   3rd Qu.:0.08000
 Max.   :1.0500    Max.   :0.35000    Max.   :0.0100000   Max.   :0.15000

   chlEpime          carotBu            carotBe           carotBB
 Min.   :0.0000    Min.   :0.0000000   Min.   :0.00000   Min.   :0.00000
 1st Qu.:0.0100    1st Qu.:0.0000000   1st Qu.:0.00000   1st Qu.:0.02000
 Median :0.0200    Median :0.0000000   Median :0.00000   Median :0.06000
 Mean   :0.0229    Mean   :0.0003226   Mean   :0.02548   Mean   :0.08355
 3rd Qu.:0.0350    3rd Qu.:0.0000000   3rd Qu.:0.00000   3rd Qu.:0.09500
 Max.   :0.0600    Max.   :0.0100000   Max.   :0.28000   Max.   :0.36000
```

El resumen consiste en un recuento o tabla de frecuencias para las variables cualitativas o factores (en este caso solamente la primera variable, "clase"), y algunos estadísticos (mínimo, primer cuartil, mediana, media, tercer cuartil, máximo) para las cuantitativas.

Para aplicar el método k-medias utilizamos la orden kmeans como se indica a continuación. La función set.seed con un número cualquiera a continuación sirve para iniciar el generador de números aleatorios, de forma que los resultados obtenidos al repetir la ejecución sean exactamente los mismos (la muestra aleatoria será la misma). Si no se utiliza, el arranque será aleatorio, y el resultado podrá ser (ligeramente) distinto

cada vez que se ejecute. En la función kmeans es aleatoria la elección de los centros iniciales de clase.

```
set.seed(12345)
modelo <- kmeans(subset(algas, select = -clase), centers = 6)
```

El conjunto de datos al que aplicamos el algoritmo consiste en todas las variables excepto "clase", para lo cual utilizamos la función subset (el signo menos indica que se suprime la variable "clase"). Tambien podríamos hacerlo de otro modo indicando las variables a utilizar (desde la 2 hasta la 20):

```
set.seed(12345)
modelo <- kmeans(algas[,2:20], centers = 6)
```

Se ha creado un objeto "modelo" con los resultados obtenidos. Si ejecutamos el nombre del objeto (marcamos "modelo" y pulsamos ejecutar) veremos todos los resultados:

```
> modelo
```

Indica en primer lugar el número de elementos para cada una de las 6 clases construidas:

```
K-means clustering with 6 clusters of sizes 4, 5, 4, 4, 8, 6
```

Los centros de las 6 clases:

```
Cluster means:
   chloroC2  chloroC1   methyl_c peridin fucoxant neoxant violaxan diadinox
1 0.0000000 0.0000000 0.02500000   0.000    0.000  0.0000   0.6425    0.000
2 0.8860000 0.0000000 0.00000000   1.196    0.000  0.0000   0.0000    0.138
3 0.4975000 0.0000000 0.00000000   0.000    0.000  0.0000   0.0000    0.000
4 0.0000000 0.0000000 0.00000000   0.000    0.000  0.0000   0.0000    0.000
5 0.0000000 0.0000000 0.00375000   0.000    0.000  0.1425   0.0625    0.000
6 0.4633333 0.3783333 0.04166667   0.000    1.435  0.0000   0.0000    0.000

     dinoxant alloxant zeaxanth lutein chloropB chl_b_epr chlAllome
1 0.160000000   0.0000  0.10500 0.0025  0.00000   0.00000   0.09750
2 0.000000000   0.0000  0.00000 0.0000  0.00000   0.00000   0.01800
3 0.000000000   0.9425  0.00000 0.0000  0.00000   0.00000   0.04750
4 0.000000000   0.0000  0.35000 0.0000  0.00000   0.00000   0.01750
5 0.008750000   0.0000  0.06375 0.7325  0.22625   0.00375   0.03875
6 0.008333333   0.0000  0.00000 0.0000  0.00000   0.00000   0.06000

    chlEpime carotBu carotBe    carotBB
1 0.04750000  0.0000  0.0000 0.04000000
2 0.02800000  0.0000  0.0000 0.06800000
3 0.02250000  0.0025  0.1125 0.01250000
4 0.01250000  0.0000  0.0000 0.31250000
5 0.01375000  0.0000  0.0425 0.06625000
6 0.02166667  0.0000  0.0000 0.04333333
```

La clase asignada a cada elemento del conjunto:

```
Clustering vector:
1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31
6  6  6  6  6  6  5  5  5  5  5  5  5  5  2  2  2  2  2  1  1  1  1  4  4  4  4  3  3  3  3
```

La suma de cuadrados, que mide la variabilidad de cada clase, y la proporción de variabilidad explicada por la clasificación (91,7%):

```
Within cluster sum of squares by cluster:
[1] 0.420500 0.199360 0.101525 0.006425 0.506100 1.408550
 (between_SS / total_SS =  91.7 %)
```

Por último la lista de componentes del objeto resultado "modelo":

```
Available components:
[1] "cluster" "centers" "totss" "withinss" "tot.withinss" "betweenss" "size" "iter" "ifault"
```

Veamos ahora si existe alguna relación entre la clase asignada a cada elemento por el algoritmo K medias y el tipo de alga definido por la variable "clase" en la primera columna (recordemos que esta variable no se ha utilizado por el algoritmo); para ello construimos una tabla de frecuencias cruzadas con ambas variables:

table(modelo$cluster, algas$clase)

	cianofíceas	clorofíceas	criptofíceas	diatomeas	dinofíceas	eustigmatoficeas
1	0	0	0	0	0	4
2	0	0	0	0	5	0
3	0	0	4	0	0	0
4	4	0	0	0	0	0
5	0	8	0	0	0	0
6	0	0	0	6	0	0

Podemos observar una coincidencia total: cada una de las clases –definida por k medias exclusivamente por la combinación de pigmentos- contiene solamente un tipo de alga. Ahora podemos interpretar cada centro de clase como la combinación "típica" de pigmentos que define a cada uno de los tipos de alga.

K-medias es un método de clasificación no supervisada, y por lo tanto no existe un procedimiento para predecir o identificar la clase o tipo de alga de un caso nuevo a partir del modelo construido. No obstante es posible añadir al conjunto de datos casos con clase desconocida (con valor 'NA' en la variable "clase") y repetir la aplicación del método: para cada caso añadido la variable modelo$cluster indicará cual es el grupo (1,2, .. 6) al que es asignado ese caso, lo que en el ejemplo anterior serviría para reconocer el tipo de alga, ya que cada clase se corresponde con un tipo distinto.

EM (Expectation – Maximization)

EM (Esperanza–Maximización, Dempster, Laird y Rubin ,1977) es un método de clasificación probabilístico, utilizado para segmentar un conjunto de datos en k clases, mediante la estimación de la función de densidad de probabilidad para cada clase, y la construcción de la función de verosimilitud, que indica la probabilidad o densidad de probabilidad asociada a la muestra observada. La función de verosimilitud se construye como combinación lineal de k componentes, los logaritmos de las funciones de probabilidad para cada clase; los parámetros de esas funciones y los coeficientes de la combinación lineal deben ser estimados en cada iteración con los datos de la muestra. Las variables tienen que ser numéricas.

Podemos considerar este método como una extensión del anterior, K-medias. En aquel, en cada iteración del algoritmo se asignaba cada elemento a la clase con el centroide más próximo; EM lo asigna a la clase más probable, de modo que sea máxima la verosimilitud de la muestra observada. Para ello se debe suponer conocida la distribución de probabilidad, y el supuesto habitual es que esa distribución es Normal multivariante, aunque también pueden emplearse otras (t-Student, Bernoulli, Poisson,…).

El algoritmo EM empieza con valores iniciales de los parámetros, que pueden ser valores aleatorios para la media inicial de cada clase y la desviación típica obtenida con toda la muestra, y repite sucesivamente dos pasos, de forma secuencial:

Esperanza: Se obtiene el valor esperado de la función log-verosimilitud –valor medio con toda la muestra- utilizando los valores de los parámetros obtenidos en la iteración anterior (o los valores iniciales si es la primera).

Maximizacion: Se obtienen nuevos valores de los parámetros, aquellos que maximizan la función de verosimilitud, es decir aquellos valores que hacen más probable la muestra realmente observada.

El algoritmo se detiene después de un número máximo de iteraciones predeterminado, cuando se repite la clasificación, o cuando el cambio en el criterio de ajuste –el cambio en el valor de la función de verosimilitud- es menor que un umbral definido a priori, obteniendo finalmente un conjunto de clases que agrupan a los elementos de la muestra, y un conjunto de funciones de probabilidad que permiten asignar elementos nuevos, no utilizados en el análisis.

La función Mclust (del paquete de R mclust) utiliza como valores iniciales los obtenidos automáticamente mediante un análisis cluster jerárquico (árbol mínimo), y explora todas las soluciones desde 1 hasta 9 clases, seleccionando finalmente el modelo óptimo -el número de clases óptimo- mediante el criterio BIC (Criterio de Información Bayesiano), que elige el modelo con mejor grado de ajuste penalizando la complejidad. Por lo tanto no es necesario indicar a priori el número de clases, aunque también puede utilizarse para un número de clases predefinido.

Debe indicarse un conjunto de datos x (data frame o matriz):

```
Mclust(x)        #  encuentra todas las clasificaciones desde 1 hasta 10 clases.
Mclust(x, G)     #  se puede indicar opcionalmente el número de clases G.
```

Utilizaremos como ejemplo el conjunto de datos "vinos", en el archivo de R "vinos.RData". Se trata de una muestra de 54 vinos monovarietales en los que se ha determinado la concentración de 6 ácidos orgánicos: galacturónico, tartárico, málico, shiquímico, cítrico, y succínico. La variedad de uva en 35 de ellos es Albariño (A), y en los 19 restantes Godello (G). Queremos saber si existe una clasificación natural de los vinos basada en las concentraciones de ácidos orgánicos, y si esa clasificación está relacionada con la variedad de uva utilizada en su elaboración.

Leemos los datos con la opción de R Commander Datos → cargar conjunto de datos, o bien ejecutando directamente la orden:

```
load("vinos.RData")     #  lee el conjunto "vinos"
```

```
> names(vinos)

[1] "var" "gal" "tar" "mal" "shi" "cit" "suc"
```

```
> summary(vinos)
 var       gal          tar          mal          shi          cit          suc
A:35   Min.:0.000   Min.:0.000   Min.:0.000   Min.:0.000   Min.:0.000   Min.:0.000
G:19   1st Q:0.455  1st Q:1.555  1st Q:1.054  1st Q:0.019  1st Q:0.101  1st Q:0.235
       Median:0.569 Median:1.988 Median:2.503 Median:0.028 Median:0.141 Median:0.311
       Mean:0.578   Mean:1.936   Mean:2.193   Mean:0.025   Mean:0.151   Mean:0.350
       3rd Q:0.684  3rd Q:2.394  3rd Q:3.291  3rd Q:0.033  3rd Q:0.177  3rd Q:0.466
       Max.:1.098   Max.:3.90    Max.:4.891   Max.:0.060   Max.:0.370   Max.:0.763
```

Aplicaremos la función Mclust, que realiza el algoritmo EM, al conjunto "vinos" excluyendo la variable "var" que indica la variedad de uva. Indicamos con el argumento G que queremos una clasificación con dos clases.

Debemos cargar el paquete mclust al que pertenece la función Mclust, con la función library, y ese paquete debe ser instalado previamente la primera vez; la instalación de un paquete es permanente, pero debe ser cargado en cada nueva sesión de R en la que se quiera utilizar.

```
library(mclust)   #  carga el paquete mclust, que debe ser instalado previamente.
modelo <-  Mclust(subset(vinos, select = -var), G=2)    # 2 grupos
```

El objeto al que hemos llamado "modelo" contiene los resultados. El de mayor interés es "classification", variable que contiene la clase –indicada por números correlativos- asignada a cada elemento de la muestra. Podemos ver si hay relación entre las dos clases creadas y la variedad de uva cruzando ambas variables:

table(modelo$classification, vinos$var)

```
   A  G
1 10 17
2 25  2
```

Vemos que en la clase 1 predominan claramente los vinos de la variedad Godello y en la clase 2 son mayoría los de la variedad Albariño, aunque 10 vinos elaborados con esta variedad han sido asignados a la clase 1. Existe una clara relación entre la clasificación "natural" encontrada por el algoritmo EM y la variedad de uva, pero esta relación no parece suficiente para identificar con precisión la variedad –sobre todo Albariño- a partir de las concentraciones de ácidos orgánicos.

La variable de resultado modelo$classification indica la clase a la que es asignado cada elemento:

modelo$classification

```
 1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27
 1  1  1  1  1  1  1  1  1  1  1  1  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2
28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54
 2  2  2  2  2  2  1  1  2  2  1  2  1  1  1  1  1  1  1  1  2  1  2  1  2  1  1
```

La variable de resultado modelo$z contiene las probabilidades de asignación a cada una de las clases, midiendo de este modo la fiabilidad de la asignación; podemos indicar el número de decimales, por ejemplo tres, con la función round:

round(modelo$z, 3)

```
    [,1]  [,2]
1  1.000 0.000
2  0.778 0.222
3  1.000 0.000
4  1.000 0.000
5  0.999 0.001
6  0.958 0.042
...  ...   ...
```

El primer caso es asignado con seguridad a la primera clase, pero el segundo, asignado a la misma clase, tiene una probabilidad 0,222 de pertenecer a la clase 2. Si algunos casos de la muestra tuviesen variedad de uva desconocida (valor NA), el método nos diría igualmente a que clase (1 o 2) han sido asignados, y de este modo la variedad de uva más probable (godello en la clase 1, albariño en la 2), aunque para este tipo de identificación es preferible utilizar los métodos de clasificación supervisada de las unidades siguientes.