Distancias

Distancias

Sitio:	eGela 2017-18 UPV/EHU
Curso:	Exploración y análisis de datos
Libro:	Distancias
Imprimido por:	JESUS MARIA YURRAMENDI MENDIZABAL
Día:	martes, 26 de septiembre de 2017, 14:09

Tabla de contenidos

1 Análisis de conglomerados (cluster analysis)

Términos similares a *clustering*:

- *Análisis de conglomerados* (cluster analysis, pattern recognition)
- *Clasificación automática* (análisis de datos)
- *Clasificación no supervisada* (estadística, análisis de datos)
- *Aprendizaje no supervisado* (inteligencia artificial)
- *Segmentación* (marketing, análisis de señales)
- *Cuantificación vectorial* (análisis de señales)
- *Ordenación* (psicología)
- *Taxonomía numérica* (biología)
- *Análisis tipológico*

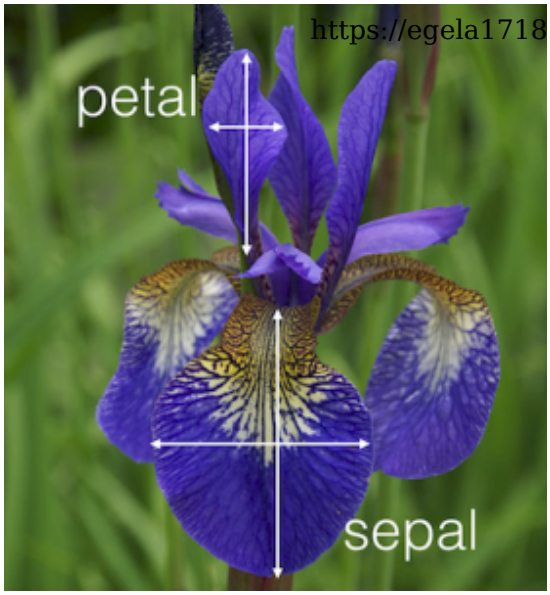
Ejemplos de áreas de aplicación:

- *Biología*: clasificación de plantas o animales a partir de sus características
- *Marketing*: grupos de clientes con comportamientos parecidos de compra
- *Planificación urbana*: identificación de grupos de viviendas de acuerdo a sus características, precios y localizaciones
- *Visión artificial*: segmentación de imágenes
- *Bioinformática*: grupos de genes y funcionalidades
- *WWW*: clasificación de documentos. Descubrimiento de patrones de acceso a Internet

El punto de partida es un conjunto de objetos sobre los que se han observado diferentes variables.

Ω	X_1		X_j		X_p
ω_1	x_{11}		x_{1j}		x_{1p}
ω_i	x_{i1}		x_{ij}		x_{ip}
ω_n	x_{n1}		x_{nj}		x_{np}

Ejemplo ('iris'):



Ω	Sepal.Length	Sepal.Width	Petal.Length	Petal.Width
1	5.1	3.5	1.4	0.2
2	4.9	3.0	1.4	0.2
3	4.7	3.2	1.3	0.2
4	4.6	3.1	1.5	0.2
5	5.0	3.6	1.4	0.2
6	5.4	3.9	1.7	0.4
...	...	...	...	...
145	6.7	3.3	5.7	2.5
146	6.7	3.0	5.2	2.3
147	6.3	2.5	5.0	1.9
148	6.5	3.0	5.2	2.0
149	6.2	3.4	5.4	2.3
150	5.9	3.0	5.1	1.8

Objetivo principal:

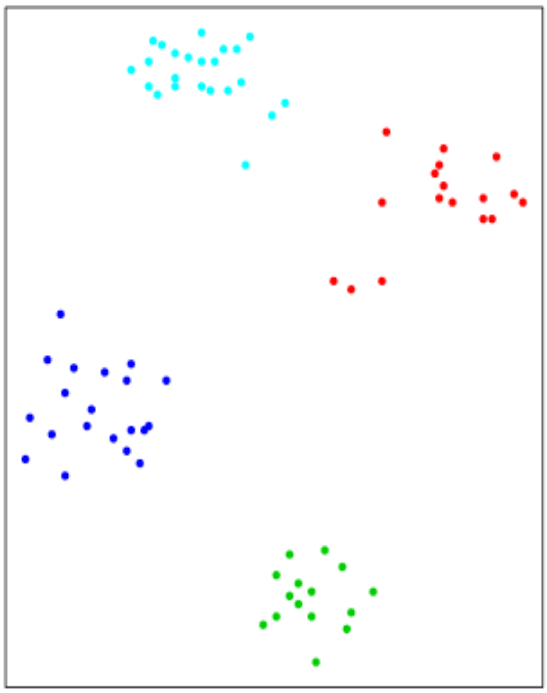
Encontrar los posibles diferentes grupos o *clusters* que forman estos objetos.

- Se trata de encontrar una *estructura* en el conjunto de datos, es decir, un *conjunto de relaciones* entre los objetos, entre las variables, y entre los objetos y las variables, que dé una información acerca del problema tratado.
- Un *cluster* es un subconjunto de objetos que son similares entre sí, y diferentes de los objetos agrupados en otros *clusters*.
- Un conjunto de *clusters* es una *estructura*.

Los *clusters* son hallados de forma *natural*, sin emplear ningún tipo de información exterior; son sugeridos por la *esencia* misma de los datos.

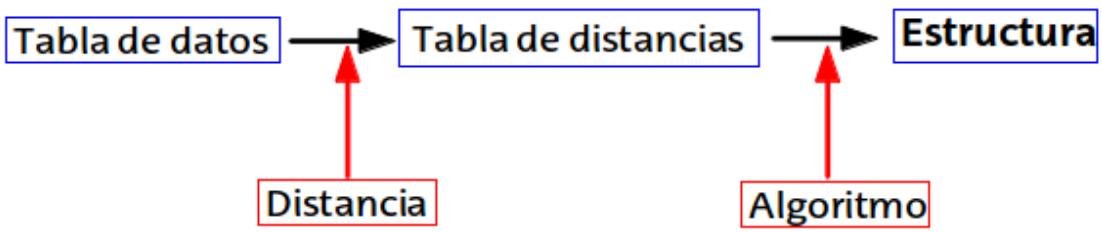
Representación geométrica de un conjunto de objetos: una variable es una dimensión.

Antes de agrupar



Después de agrupar

- Se trata de buscar *clusters* naturales y de describir sus propiedades (desconocidas, relacionándolas con los otros *clusters* y las variables), o de buscar agrupamientos útiles y convenientes, para un determinado objetivo, o, de buscar simplemente objetos extraños en relación a los demás.
- Se trata de transcribir sobre un computador, mediante un algoritmo, la capacidad que tenemos los humanos para agrupar en dos o tres dimensiones, y luego extenderlo a más dimensiones. Al principio parece una tarea fácil, pero en la práctica resulta difícil.
- Se necesita precisar la noción de *similitud* o *semejanza*, en base a las variables observadas.
- La noción de *desemejanza* se transcribe con la noción de *distancia*: dos objetos que son *desemejantes* son *distantes* entre sí.
- Dos o más objetos pertenecen al mismo *cluster* si son *similares*, es decir, si la *distancia* entre ellos es *pequeña*.
- Es preciso definir qué *distancia* (geométrica).



- ¿Qué es un buen conjunto de *clusters* o *clustering*? ¿Cuál es la calidad del resultado?
- Se necesita un *criterio*, y se puede tratar de buscar el *mejor clustering* respecto a ese criterio.
- No hay forma de determinar cuál es el mejor *criterio*. En función del tipo de problema cada analista determina por experimentación un criterio (o varios) del que obtiene una buena solución (o varias), si no la mejor.

- Una vez determinado un criterio, y encontrada una buena solución para el criterio, hay que determinar si la solución es válida para el problema planteado.

La calidad de un resultado es alta cuando:

- La semejanza entre los objetos dentro de una misma clase (*intra-cluster*) es alta
- La semejanza entre los objetos de clases distintas (*inter-cluster*) es baja
- La distancia entre los objetos dentro de una misma clase (*intra-cluster*) es pequeña
- La distancia entre los objetos de clases distintas (*inter-cluster*) es grande

La calidad de un resultado depende de:

- La calidad del conjunto de datos y de su preprocesamiento
- La medida de desemejanza entre objetos (distancia) que se haya elegido
- El algoritmo de clasificación (*clustering*) que se haya utilizado

Resumen

- Conjunto de datos: tabla de objetos-variables-valores
- Representación geométrica de objetos
- Definición de una distancia entre los objetos
- Definición de un criterio de bondad
- Definición de un algoritmo de *clustering*
- Validación de los *clusters* obtenidos

Direcciones complementarias

http://en.wikipedia.org/wiki/Cluster_analysis

http://en.wikibooks.org/wiki/Data_Mining_Algorithms_In_R/Clustering

http://cran.r-project.org/web/views/Cluster.html

2 Distancias

Distancias (enlazado a un documento pdf)

2.1 Distancias.R

```
#########################################################################
#
# DISTANCIAS
#
#########################################################################
#
# Introducción de los datos: 'iris'
#
?iris
#
# Disimilitud o desemejanza entre objetos (plantas): distancia
#
# 4 variables cuantitativas y una cualitativa (5a.variable)
#
# NO se toma en cuenta la variable 'Species' de la planta (5a. variable)
# porque el objetivo es formar clusters de plantas
# sin conocer la clase a la que pertenecen,
# de forma que 'a posteriori' se pueda verificar que
# las clases encontradas se corresponden, en términos generales,
# con las clases etiquetadas.
#
# No es una situación real,
# pero ayuda a mostrar la fortaleza de estas técnicas.
#
# Es un ejemplo clásico 'de salón' (o 'de juguete').
#
datos0 <- iris[,1:4]
dim(datos0); nrow(datos0); ncol(datos0)
head(datos0)
summary(datos0)
#
# Preprocesamiento
#
# ¿Estandarizar las variables?
#
# Las variables son homogéneas (cm.); es razonable tratarlas tal cual.
#
# Sin embargo, algunas medidas son más grandes ('*.Length') que otras,
# y tratarlas tal cual equivaldría, en cierto sentido,
# a analizarlas por su apariencia visual, y, puede ser,
# que éste no sea el punto de vista deseado.
#
# Estandarizar las variables (media 0, varianza 1) equivale, en principio,
# a poner todas las variables a un mismo nivel de incidencia en el
análisis.
#
# La estandarización de las variables es un tipo de preprocesamiento.
#
# Para estandarizar las variables: 'scale()'
```

```
datosz <- scale(datos0)
#
# Verificación:
#
for(j in 1:ncol(datosz)) print(c(mean(datosz[,j]), var(datosz[,j])))
#
# En adelante los datos serán los estandarizados.
# Es la opción que se hace en el siguiente análisis.
#
datos <- datosz
#
# si no lo fueran:
# datos <- datos0
#
#########################################################################
#
# Cálculo de las distancias entre los elementos: 'dist()'
#
# http://es.wikipedia.org/wiki/Distancia
#
# http://en.wikipedia.org/wiki/Distance
#
# Ejecuta el siguiente comando para vislumbrar las prestaciones de
'dist()',
# que serán detalladas más adelante
#
example(dist)
#
?dist
#
# Distancia euclidiana:
#
distancias <- dist(datos, method="euclidean")
#
# Características generales del objeto 'distancias'
#
class(distancias)
mode(distancias)
attributes(distancias)
str(distancias)
#
#########################################################################
#
# Cuestiones importantes en el cálculo de las distancias.
#
# A. Estandarización de los datos
#
# B. Método de la distancia
#
# ###########
#
```

```
# A. Importancia de la cuestión de la estandarización
#
par(mfrow=c(2,2)) # cuatro gráficos en uno
#
# Dos gráficos para datos estandarizados (1)
# Dos gráficos para datos originales, no estandarizados (2)
#
# 1. Datos estandarizados
#
datos <- datosz
#
distancias <- dist(datos, method="euclidean")
#
# Visualización de las distancias resultantes
#
hist(distancias, main="Datos estandarizados")
#
# Mapa de calor para distancias por pares de objetos:
# A menor distancia, mayor intensidad de calor (color rojo)
#
image(as.matrix(distancias), col=heat.colors(12), axes=FALSE,
      xlab="Objetos", ylab="Objetos",
      main="A menor distancia,\nmayor intensidad")
#
# 2. Datos originales, no estandarizados
#
datos <- datos0
#
distancias <- dist(datos, method="euclidean")
#
hist(distancias, main="Datos originales")
#
# Mapa de calor
#
image(as.matrix(distancias), col=heat.colors(12), axes=FALSE,
      xlab="Objetos", ylab="Objetos",
      main="A menor distancia,\nmayor intensidad")
#
# Comentario:
# El data.frame 'iris' está originalmente ordenado
# por clases de plantas (primero las 50 'setosa',
# luego las 50 'versicolor', y, finalmente, las 50 'virginica').
# Por eso resulta estructurada la imagen.
#
# En una aplicación real los objetos están 'desordenados'
# y se plantea el problema de 'ordenarlos', de forma que
# la imagen refleje la estructura de clases.
#
# Conclusión:
# La distribución de los valores de las distancias
# en (1) y en (2) es distinta.
#
```

```
# Las imágenes son bastante similares. No es siempre así.
#
par(mfrow=c(1,1)) # valores iniciales
#
############
#
# B. Importancia de la cuestión del método de la distancia #
par(mfrow=c(3,2)) # seis gráficos en uno
#
par(mfrow=c(3,2)) # seis gráficos en uno
#
# Dos gráficos para method="manhattan" (1)
# Dos gráficos para method="euclidean" (2)
# Dos gráficos para method="maximum" (3)
#
# 1. method= "manhattan"
#
datos <- datosz
#
distancias <- dist(datos, method="manhattan")
#
hist(distancias, main="Datos estandarizados")
#
image(as.matrix(distancias^2), col=heat.colors(12), axes=FALSE,
      xlab="Objetos", ylab="Objetos", main="Método=Manhattan")
#
# 2. method="euclidean"
#
datos <- datosz
#
distancias <- dist(datos, method="euclidean")
#
hist(distancias, main="Datos estandarizados")
#
image(as.matrix(distancias), col=heat.colors(12), axes=FALSE,
      xlab="Objetos", ylab="Objetos", main="Método=Euclidean")
#
#
# 3. method="maximum"
#
datos <- datosz
#
distancias <- dist(datos, method="maximum")
#
hist(distancias, main="Datos estandarizados")
#
image(as.matrix(distancias^2), col=heat.colors(12), axes=FALSE,
      xlab="Objetos", ylab="Objetos", main="Método=Maximum")
#
# Conclusión:
# La distribución de los valores en (1), en (2) y en (3) es distinta
# Las imágenes son distintas. En una hay más contraste que en las otras.
```

```
#
par(mfrow=c(1,1)) # valores iniciales
#
#########################################################################
#
# A partir de los métodos ofrecidos por 'dist()'
# pueden definirse otras distancias
#
# Ejemplo: distancia euclidiana cuadrática
#
# Las distancias euclidianas cuadráticas están en consonancia
# con la noción de varianza
#
# Recordatorio: fórmulas de la varianza
#
# xvar <- sum((x-mean(x))^2))/length(x)
#
# xvar.3 <- sum(dist(x)^2)/length(x)^2
#
# xvar.3 sin referencia a la media
#
# xvar.3 == xvar
#
# Verificación en el caso multivariante
#
datos <- datosz
distancias <- dist(datos, method="euclidean")
#
# Suma de distancias cuadráticas (Sum of Square) respecto a la media,
'SST'
#
SST <- 0
for(j in 1:ncol(datos)) SST <- SST + (nrow(datos)-1)*var(datos[,j])
SST
#
c(sum(distancias^2)/nrow(datos)^2, SST/nrow(datos))
sum(distancias^2)/nrow(datos)^2 == SST/nrow(datos)
#
# La selección del tipo de distancia es una decisión importante,
# pues, en general, se obtienen resultados distintos.
#
par(mfrow=c(2,2)) # cuatro gráficos en uno
#
# 1. method = "euclidean"
#
hist(distancias, main="Datos estandarizados")
#
image(as.matrix(distancias), col=heat.colors(12), axes=FALSE,
xlab="Objetos", ylab="Objetos", main="Método=Euclidean")
#
# 2. method = "squared euclidean"
#
```

```
distancias2 <- distancias^2
#
hist(distancias2, main="Datos estandarizados")
#
image(as.matrix(distancias2), col=heat.colors(12), axes=FALSE,
xlab="Objetos", ylab="Objetos", main="Método=Euclidean Cuadrático")
#
# Conclusión:
# Las distribuciones de los valores de distancias es distinta.
# Hay más contraste en una imagen que en otra.
#
# par(mfrow=c(1,1)) # valores iniciales
#
##########################################################################
#
# Distancia entre variables
# basada en el coeficiente de correlación lineal
#
# Si se considera una variable X_i en R^n, descrita por
# las coordenadas dadas por los valores estandarizados de los n objetos,
# y la distancia euclidiana cuadrática entre las variables
# afectada por el factor 1/n (en media),
# resulta: 2·(1-cor(X_i,X_j))
# Abstrayendo el factor 2,
# las variables se encuentran en la hiperesfera (R^n) de radio 1.
#
# Las distancias entre variables son longitudes de cuerdas.
# La máxima distancia es 2, la longitud del diámetro, y corresponde al
caso
# de variables correladas lineal e inversamente (cor(X_i,X_j) = -1)
#
distanciasX <- (1-cor(datos))
class(distanciasX)
#
distanciasX <- as.dist(distanciasX, diag=TRUE, upper=TRUE)
class(distanciasX)
#
round(distanciasX, digits=3)
#
# Se observa que 'Sepal.Width' está distante de las otras,
# que resultan bastante cercanas entre sí (altamente correladas)
#
##########################################################################
```

3 Estructuras

Estructuras (enlazado a un fichero *pdf*)