Repaso multivariante. Combinando PCA con clustering

00R Team

Abril 2016

Índice

1. Preliminares.

Cargar/instalar las librerías `FactoMineR, factoextra, rgl, cluster`

```
# código R
# cargar librerías
```

2. Cargar datos

1. Cargar los datos de `iris`
2. Visualizar matriz de correlaciones numérica y gráficamente:
   - `cor()`
   - `plot()`
3. Colorear los puntos según las especies

```
# código R
# data( ... )
# irisdf <- iris[,1:4]
# plot( ..., col = iris$Species, pch = 20 )
# .....( ) # matriz de correlaciones ¿?¿?¿?
```

- ¿Hay correlación entre la variables?
- Si la hay, ¿es muy alta?

3. Componentes principales con `FactoMiner`

1. Calcular componentes principales con la función `PCA()`
2. La función permite normalizar los datos "al vuelo"

```
# código R
# irisPCA <- PCA( .... , ...... , graph = FALSE )
# .... ( irisPCA ) # resumen del análisis
```

`* ¿Cuántas variables necesitamos para tener el 90% de la varianza?`

1. Visualizar correlaciones de datos en nuevas variables (componentes)
    - Ayuda: `newdfIris <- as.data.frame( irisPCA$ind$coord )`

```
# newdfIris <- as.data.frame( ...matriz factorial de los individuos.... )
# plot( .... , col = .... , pch = 20 ) # colores por especies
# ...( ..... )  # calcular correlaciones entre las nuevas variables
```

- ¿Cómo son ahora las correlaciones entre las nuevas "variables"?

3.1. biplot PCA

Hacer biplot del análisis:

```
# código R
# fviz_pca_biplot( ... ) # biplot del PCA
```

- ¿Qué interpretación le das a los nuevos ejes?

3.2. Visualizar datos en 3D

```
color3d = factor( iris[,5], labels = c(1:3) )
#color3d = 1
plot3d( newdfIris, col = color3d, type = "s", radius = .05 )
```

3.3. Biplot en 3D

```
plot3d( irisPCA$ind$coord[,1:3], type = "s", radius = 0.05,
        col = color3d, size = 1, aspect = 1)
text3d(irisPCA$var$coord[,1:3]*3, texts=rownames(irisPCA$var$coord), col="red")
coords <- NULL
for (i in 1:nrow(irisPCA$var$coord)) {
  coords <- rbind(coords, rbind(c(0,0,0),irisPCA$var$coord[i,1:3])*3)
}
lines3d(coords, col="red", lwd=4)
```

- ¿La tercera dimensión mejora la interpretación de las contribuciones de las variables?

4. Kmeans sobre PCA

4.1. Kmeans k=3

Podemos hacer una clasificación sobre los la nueva matriz de datos generada a partir del PCA. Esta matriz de datos esta contenida en la variable del `irisPCA$ind$coord`.

Vamos a hacer una clasificación no jerárquica sobre esos datos con K=3 grupos.

```
# código R
# k <- ¿?
# iriskm <- kmeans( .... , ... , .... , nstart = 3)
clusplot( newdfIris, iriskm$cluster, color = TRUE, labels = 4,
          col.p = iris$Species, lines = 0 )
```

4.2. Cluster 3d

La tercera dimensión puede ayudar a visualizar los grupos generados. Podemos comparar la distribución de los individuos original

```
open3d()
plot3d(newdfIris, type = "s", radius = 0.05,
        col = iriskm$cluster, size = 1, aspect = 1, main = "Kmeans" )
open3d()
plot3d( newdfIris, type = "s", radius = 0.05,
        col = color3d, size = 1, aspect = 1, main = "Original" )
```

Con `hclust` podemos ver cómo se relacionan esos grupos. Para 3 grupos no sería necesario porque son fáciles de identificar las relaciones. Hay un grupo mejor definido que los otros dos.

Para esto se puede hacer un clustering jerárquico de los centroides de los grupos como representantes de éstos. La variable que contiene estos datos es `iriskm$centers`; por tanto es de este data.frame del que hay que calcular la matriz de distancias y el dendrograma.

5. Clasificación jerárquica sobre centroides de `kmeans`

```
# Código R
iriskm$centers
# hdist <- ....( ..... ) # matriz de distancias de los centroides
# irishcl <- .....( .... , method = "ward.D2" ) # clustering jerárquico
# ....( .... , hang = -1 ) # visualizar dendrograma
```

5.1. Cortar dendrograma en 3 grupos

Cortar en 3 grupos un dendrograma de 3 elementos puede parecer que no tiene sentido, pero nos va a permitir etiquetar a los individuos según estos grupos. Cuando lo hagamos para un `kmeans` con más grupos veremos su sentido.

1. Hacer el corte con la función `cutree()` y asignar los grupos a una nueva variable

```
# corte <- .... ( irishcl, k = 3 )
# corte
#iriskm$cluster
mycluster <-  iriskm$cluster
```

Con `table()` podemos comprobar cómo se han clasificado los individuos frente a la situación original

```
iris$newgroup <- as.factor( mycluster )
( tabla3grupos <- table( iris$Species, iris$newgroup,
                         dnn = c("Originales", "PCA_Kmeans_3_Hclust" )) )
```

```
* ¿Es una buena clasificación?

* Respecto a los datos originales, ¿hay mucho error al clasificar los individuos?

Podemos comparar la distribución de los individuos según la especie
a la que pertenecen frente a la clasificación hecha con `kmeans`
sobre componentes principales
```

```
par( mfrow = c( 1, 2 ) )
# ....( .... , col = iris$Species, main = "Sólo PCA", asp = 1 )
# ....( .... , col = iris$newgroup,
#     main = "PCA + Kmeans K=3 + Hclust = 3", asp =1 )
par( mfrow = c( 1, 1 ) )
```

- Desde un punto de vista estadístico ¿Es una buena clasificación?
- ¿Podemos mejorar está clasificación?

5.2. Kmeans $k = n$ y corte del árbol en n_2 grupos

Vamos a probar `kmeans` con k=6. El código es el mismo que en el caso de k=3. Salvo una pequeña recodificación que hay que hacer al final.

```
## código R
##kmeans 6 grupos

# k <- 6
# iriskm <- kmeans( ... , ... )
# clusplot( .... )

##hclust

iriskm$centers
# hdist <- ...( ... ) # calcular distancias de los centroides
# irishcl <- ....( ... , method = "ward.D2" )
# ....( .... , hang = -1 ) #visualizar dendrograma

# Cortar dendrograma en 3 grupos
# corte <- .... ( .... , k = ... )
corte
```

```
codigoGrupo <- corte
myclusterCortado <- codigoGrupo[ iriskm$cluster ]

plot( irisPCA$ind$coord[ , 1:2 ], type = "n", asp = 1 )
text( irisPCA$ind$coord[ , 1:2 ], as.character( myclusterCortado ), col = iriskm$cl )

## Comparar con los datos originales.
iris$newgroup2 <- as.factor( mycluster )
( tabla6grupos <- table( iris$Species, iris$newgroup2,
                         dnn = c("Originales", "PCA_Kmeans_6_Hclust" ) ) )
```

6. La función HCPC

Con la función `HCPC()` del paquete `FactoMineR` permite hacer una clasificación directamente sobre el resultado de un análisis de factores tal como un componentes principales.

```
irisHCPC <- HCPC( PCA(irisdf, scale.unit = TRUE, graph = FALSE), graph = FALSE )
plot( irisHCPC, choice = "3D.map" )
fviz_cluster( irisHCPC )
irisHCPC$data.clust
( tablaHCPC <- table ( iris$Species, irisHCPC$data.clust$clust,
                       dnn = c("Originales", "PCA + HCPC" ) ) )
```

7. Comparando resultados

Finalmente podemos comparar las clasificación realizadas por los 3 métodos. Simplemente hay que visualizar las 3 tablas de contingencia.

```
# código R
# ...
```

- ¿Hay algún método mejor que otro?
- ¿Hay alguna cosa que llame la atención?