Estadística Aplicada III

Métodos de agrupación: Escalamiento Multidimensional

Jorge de la Vega Góngora

Departamento de Estadística,
Instituto Tecnológico Autónomo de México

Semana 15

Introducción

Introducción

- Escalamiento multidimensional (MDS) es un conjunto de algoritmos que resuelven el problema de representar objetos espacialmente, a través de construir una configuración de puntos en alguna dimensión menor en $\mathbb{R}^k$, para $k = 1, 2, 3$, utilizando la información disponible sobre la *similitud* o *disimilitud* de los objetos, de tal manera que las proximidades de los items en esta representación, se 'parezcan' lo más posible a las similitudes o disimilitudes originales.
- En este contexto se introduce una medida de cercanía llamada *stress*, para medir cómo la configuración "ajustada" se apega a la configuración "real".
- MDS también se considera una técnica exploratoria de análisis multivariado, así como una técnica de reducción de dimensión. Fue creado originalmente en 1952 por Warren S. Torgeson y posteriormente desarollado y extendido por Joseph Kruskal.

Ejemplos de aplicación MDS

- Piscología: Mapa perceptual de estímulos psicológicos
- Mercadotecnia: Mapas de productos de elección de consumidor y selección de productos
- Ecología: mapas de impacto ambiental de contaminación
- Redes sociales: Gráficas de llamadas telefónicas. Vértices son múmeros de teléfono y arcos son llamadas entre ellos.
- Música: Usar una medida de calidad musical del sonido como entrada a una medida de distancia lineal para evaluar las similitudes y diferencias entre ina variedad de canciones.

Tipos de soluciones en MDS I

En multiescalamiento dimensional (MDS) hay dos tipos posibles de soluciones:

- **MDS-métrico o clásico**: se utilizan las similitudes (o distancias) originales para obtener una representación geométrica en k dimensiones. Esta versión también se conoce como análisis de coordenadas principales.
  - No hay una solución única para la representación geométrica, porque la solución es invariante a rotaciones, reflexiones o traslaciones. Usualmente el problema de traslación se resuelve ubicando el vector de medias en el origen.
- **MDS-no métrico**: cuando sólo se utiliza información ordinal (basada en los rangos) de las similitudes originales.

Similitud/disimilitud

Características generales de la similitud I

- El concepto de similitud es bastante general y puede incluso ser subjetiva. Se puede definir para varios tipos de datos: cuantitativos, binarios, nominales ordinales o mixtos.

Ejemplos

- La presencia o ausencia de ciertas características se pueden usar como medida de similitud: los objetos serán más similares si comparten más características
- puede ser una medida de asociación, como una correlación o alguna medición de frecuencia de confusión (qué tanto se confunde con otro en una identificación, etc).
- 12 marcas de yogurth evaluadas por 10 jueces en nueve variables. Los yogurths son presentados en pares a los panelistas a los que se les pide evaluar que tan similares son las dos muestras en una linea de escala descriptiva de 15cm.
- La similitud entre códigos Morse puede medirse como el porcentaje de veces que las personas confunden las sucesiones de símbolos después de escucharlos en una sucesión rápida.

- Una función de similitud puede ser simétrica, no negativa y creciente conforme los objetos son más similares.
- Se considera que una medida de similitud es inversamente proporcional a una medida de distancia. La distancia puede ser considerada como una medida de disimilitud.

Definición

- Una *matriz de similitud* **C** es simétrica ($\mathbf{C}' = \mathbf{C}$) y tal que

$$0 \leq c_{ij} \leq c_{ii} \quad \forall i, j$$

- Una *matriz de disimilitud* o *distancia* **D** también es simétrica y

$$d_{ii} = 0, \quad d_{ij} \geq 0 \quad i \neq j.$$

- Con frecuencia se intercambian los coeficientes de similitud a distancia y viceversa. Posibles transformaciones incluyen:
  - $d_{ij} = c - c_{ij}$ para alguna constante c.
  - $c_{ij} = \frac{1}{1+d_{ij}}$
  - La transformación estándar: $d_{ij} = (c_{ii} - 2c_{ij} + c_{jj})^{1/2}$

Ejemplos I

A continuación consideraremos varios ejemplos de medidas, tomando en cuenta el tipo de variable (discreta, continua, binaria) y las escalas de medición (nominal, ordinal, de intervalo, de razón). Algunas de estas ya las hemos definido antes:

- Continuas:
  - **Distancia Euclideana:** La distancia usual para variables numéricas:
$$d(\mathbf{x}, \mathbf{y}) = \sqrt{(\mathbf{x}-\mathbf{y})'(\mathbf{x}-\mathbf{y})}$$
  - **Distancia de Mahalanobis:** Los datos se ponderan por su variabilidad:
$$d(\mathbf{x}, \mathbf{y}) = \sqrt{(\mathbf{x}-\mathbf{y})'\mathbf{S}^{-1}(\mathbf{x}-\mathbf{y})},$$
aunque no siempre se conocen los grupos de antemano y por lo tanto no se puede estimar **S** (como en conglomerados).
  - **Norma supremo:**
$$d(\mathbf{x}, \mathbf{y}) = max|x_i - y_i|$$
  - **Distancia de Minkowski:**
$$d_m(\mathbf{x}, \mathbf{y}) = \left[\sum_{i=1}^{p} |x_i - y_i|^m\right]^{1/m}.$$
Cuando $m = 1$ es la 'distancia Manhattan'.
- No negativas:

Ejemplos II

- **métrica de Canberra:**

$$d(\mathbf{x}, \mathbf{y}) = \sum_{i=1}^{p} \frac{|x_i - y_i|}{x_i + y_i}$$

- **coeficiente de Czekanowski:**

$$d(\mathbf{x}, \mathbf{y}) = 1 - \frac{2 \sum_{i=1}^{p} \text{mín}(x_i, y_i)}{\sum_{i=1}^{p} x_i + y_i}$$

- Binarias:
  - **Presencia o ausencia de características:**

$$\sum_{i=1}^{p} (x_{ij} - x_{kj})^2,$$

donde:

$$(x_{ij} - x_{kj})^2 = \begin{cases} 0 & x_{ij} = x_{kj} = 1 \text{ o } x_{ij} = x_{kj} = 0 \\ 1 & x_{ij} \neq x_{kj} \end{cases}$$

aquí estamos comparando la $i-$ésima variable de los items j y k.

Fórmula de Gower I

- Siempre es posible construir similaridades a partir de distancias, con las transformaciones mencionadas antes.
- Sin embargo, disimilitudes que son distancias reales no siempre pueden ser construidas a partir de similitudes. Esto sólo se puede hacer si la matriz **C** es definida positiva (Gower, 1971).
- Con la condición anterior, y con la similitud máxima escalada de tal forma que $\tilde{c}_{ii} = 1$,

$$d_{ik} = \sqrt{2(1 - \tilde{c}_{ik})}$$

Esta es la fórmula de Gower, que tiene propiedades de distancia.

Funciones en R para distancias I

- En R hay algunas funciones para calcular matrices de distancias a partir de datos:
  - la función `dist` que puede calcular a partir de una matriz numérica o data.frame las distancias: euclidean, max, manhattan, canberra, binary o minkoswi:

```
x <- matrix(rnorm(100),nrow=5)
dist(x) #euclideana por default
          1        2        3        4
2 5.420849
3 6.432524 4.355257
4 5.956440 7.086266 6.492044
5 5.229745 5.601363 6.181926 6.872347
dist(x,"canberra")
          1        2        3        4
2 15.00031
3 17.21744 10.52925
4 14.43011 14.44017 14.90150
5 14.27737 15.70712 14.79091 15.21812
dist(x,"binary") #revisar definición de binary
  1 2 3 4
2 0
3 0 0
4 0 0 0
5 0 0 0 0
```

Funciones en R para distancias II

- La función `daisy` que calcula matrices de disimilaridades en donde las variables pueden ser de tipos mezclados. En este caso, aplica una generalización de la transformación de Gower que se mencionó arriba:

Funciones en R para distancias III

```r
library(cluster)
data(flower) #características de 18 flores,
str(flower)
```

```
'data.frame': 18 obs. of  8 variables:
 $ V1: Factor w/ 2 levels "0","1": 1 2 1 1 1 1 1 1 2 2 ...
 $ V2: Factor w/ 2 levels "0","1": 2 1 2 1 2 2 1 1 2 2 ...
 $ V3: Factor w/ 2 levels "0","1": 2 1 1 2 1 1 1 2 1 1 ...
 $ V4: Factor w/ 5 levels "1","2","3","4",..: 4 2 3 4 5 4 4 2 3 5 ...
 $ V5: Ord.factor w/ 3 levels "1"<"2"<"3": 3 1 3 2 2 3 3 2 1 2 ...
 $ V6: Ord.factor w/ 18 levels "1"<"2"<"3"<"4"<..: 15 3 1 16 2 12 13 7 4 14 ...
 $ V7: num  25 150 150 125 20 50 40 100 25 100 ...
 $ V8: num  15 50 50 50 15 40 20 15 15 60 ...
```

```r
round(daisy(flower,metric="gower"),2)
```

```
Dissimilarities :
      1    2    3    4    5    6    7    8    9   10   11   12   13   14   15   16   17
2  0.89
3  0.53 0.51
4  0.35 0.55 0.57
5  0.41 0.62 0.37 0.64
6  0.23 0.66 0.30 0.42 0.34
7  0.29 0.60 0.49 0.34 0.42 0.19
8  0.42 0.46 0.60 0.30 0.47 0.57 0.41
9  0.58 0.43 0.45 0.81 0.33 0.51 0.59 0.64
10 0.61 0.45 0.47 0.56 0.38 0.41 0.59 0.66 0.43
11 0.33 0.71 0.60 0.65 0.39 0.48 0.57 0.50 0.43 0.39
12 0.43 0.59 0.60 0.51 0.50 0.52 0.64 0.42 0.42 0.38 0.26
13 0.52 0.52 0.54 0.75 0.29 0.45 0.53 0.58 0.22 0.36 0.34 0.23
14 0.29 0.59 0.61 0.37 0.52 0.37 0.50 0.46 0.44 0.36 0.28 0.16 0.38
15 0.62 0.39 0.53 0.55 0.46 0.51 0.33 0.45 0.25 0.42 0.48 0.43 0.32 0.44
16 0.69 0.36 0.62 0.34 0.73 0.51 0.44 0.64 0.65 0.35 0.74 0.61 0.59 0.46 0.39
17 0.78 0.19 0.58 0.42 0.69 0.59 0.52 0.47 0.61 0.31 0.70 0.56 0.55 0.54 0.35 0.17
18 0.46 0.45 0.72 0.44 0.48 0.64 0.47 0.14 0.52 0.81 0.54 0.55 0.57 0.57 0.51 0.78 0.61

Metric :  mixed ;  Types = N, N, N, N, O, O, I, I
Number of objects : 18
```

Solución métrica

Planteamiento del problema

- Consideren n puntos $P_1 = \mathbf{x}_1, \ldots, P_n = \mathbf{x}_n \in \mathbb{R}^p$, correspondientes a los n renglones de la matriz de datos **X**.
- En la solución métrica se supone que la matriz de proximidad **D** es una matriz con las distancias euclídea entre todos los pares de puntos P_i y P_j.

Problema métrico o clásico

El problema consiste en encontrar la matriz de coordenadas **X** a partir de la matriz de distancias euclídeanas **D**.

- La matriz de *productos interiores* $\mathbf{B} = \mathbf{X}\mathbf{X}'$ tiene componentes $b_{ij} = \mathbf{x}_i'\mathbf{x}_j$. La solución clásica usa **D** para encontrar **B** y entonces de **B** se obtienen los puntos P_i factorizando **B** en $\mathbf{X}\mathbf{X}'$. De hecho, las distancias euclídeas entre los renglones de **X** se pueden escribir en términos de los elementos de **B** como:

$$d_{ij}^2 = b_{ii} + b_{jj} - 2b_{ij}$$

- La solución se puede obtener de manera única si se restringe a que considerar $\bar{\mathbf{x}} = \mathbf{0}$.

Solución métrica I

- Para escribir las b's en términos de las d's y considerando la restricción $\bar{\mathbf{x}} = \mathbf{0}$, se puede ver que los términos de **B** se pueden escribir como:

$$b_{ij} = -\frac{1}{2}(d_{ij}^2 - d_{i\cdot}^2 - d_{\cdot j}^2 + d_{\cdot\cdot}^2)$$

  Lo anterior es equivalente a tomar: $\mathbf{B} = \mathbf{HAH}$ con $\mathbf{H} = \mathbf{I} - \frac{1}{n}\mathbf{1}\mathbf{1}'$ y $\mathbf{A} = (-\frac{1}{2}d_{ij}^2)$.
- Construída **B** con los factores indicados, sigue su factorización. Usando la descomposición espectral, considerando que el rango es q:

$$\mathbf{B} = \mathbf{V}\mathbf{L}^{1/2}\mathbf{L}^{1/2}\mathbf{V}'$$

  donde $\mathbf{L} = \text{diag}\,(\lambda_i), \lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_p > 0$ y **V** es la matriz de los p eigenvectores normalizados.
- Entonces: $\mathbf{X} = \mathbf{V}\mathbf{L}^{1/2}$
- Podemos seleccionar el número de dimensiones como siempre tomando la proporción siguiente haciendo la gráfica de codo respectiva:

$$P_k = \frac{\sum_{i=1}^{k} \lambda_i}{\sum_{i=1}^{p} \lambda_i}$$

- Usualmente un valor del orden de 0.8 se considera en la práctica un ajuste razonable.

Equivalencia de la solución métrica a componentes principales I

- Cuando la matriz de similitudes contiene distancias euclídeas calculadas de una matriz de datos $\mathbf{X}_{n\times p}$, entonces $MDS \equiv PCA$, donde las coordenadas de los puntos $P_1, P_2, \ldots, P_n$ corresponden a los scores de las componentes principales extraídas de la matriz de covarianzas de los datos.

Dualidad entre Componentes Principales y Escalamiento Multidimensionl métrico

- PCA reduce dimensiones, preservando la covarianza entre los datos.
- MDS reduce dimensiones, preservando distancias entre puntos.
- Si $Cov(\text{datos}) = Dist(\text{datos})$, entonces $MDS \equiv PCA$

Ejemplo 1 I

Ejemplo. [Dualidad entre componentes principales y escalamiento multidimensional métrico]

En este ejemplo verificaremos la dualidad entre escalamiento multidimensional métrico y componentes principales.
Consideremos una matriz **X** de $n = 10$ puntos arbitrarios en $\mathbb{R}^5$ y construyamos su matriz de distancias euclídeas:

Ejemplo 1 II

```
X <- matrix(c(3,4,4,6,1, 5,1,1,7,3, 6,2,0,2,6, 1,1,1,0,3, 4,7,3,6,2,
              2,2,5,1,0, 0,4,1,1,1, 0,6,4,3,5, 7,6,5,1,4, 2,1,4,3,1),
              nrow=10, byrow=T)

D <- dist(X) #distancias euclídeas
D   # Matriz de distancias

          1        2        3        4        5        6        7        8        9
2  5.196152
3  8.366600 6.082763
4  7.874008 8.062258 6.324555
5  3.464102 6.557439 8.366600 9.273618
6  5.656854 8.426150 8.831761 5.291503 7.874008
7  6.557439 8.602325 8.185353 3.872983 7.416198 5.000000
8  6.164414 8.888194 8.366600 6.928203 6.000000 7.071068 5.744563
9  7.416198 9.055385 6.855655 8.888194 6.557439 7.549834 8.831761 7.416198
10 4.358899 6.164414 7.681146 4.795832 7.141428 2.645751 5.099020 6.708204 8.000000

# por ejemplo, la distancia entre P1 y P2:
sqrt(sum((c(3,4,4,6,1)-c(5,1,1,7,3))^2))

[1] 5.196152

# la distancia entre P2 y P3:
sqrt(sum((c(6,2,0,2,6)-c(5,1,1,7,3))^2))

[1] 6.082763
```

Como $p = 5$, los eigenvectores del 6 al 9 son prácticamente 0. Entonces podemos recobrar la matriz de distancia con los primeros 5 eigenvectores:

Ejemplo 1 III

```
cmdscale(D, k = 9, eig = T)

Warning in cmdscale(D, k = 9, eig = T): only 7 of the first 9 eigenvalues are > 0

$points
           [,1]        [,2]       [,3]       [,4]        [,5]          [,6]         [,7]
 [1,] -1.6038325 -2.38060903  2.2301092 -0.3656856  0.11536476  2.153241e-08 0.000000e+00
 [2,] -2.8246377  2.30937202  3.9523782  0.3419185  0.33169405 -6.457153e-10 3.627900e-09
 [3,] -1.6908272  5.13970089 -1.2880306  0.6503227 -0.05133897  3.035403e-09 7.346907e-09
 [4,]  3.9527719  2.43233961 -0.3833746  0.6863995 -0.03460933  8.968172e-09 2.632470e-09
 [5,] -3.5984894 -2.75538195  0.2551393  1.0783741 -1.26125237 -5.196705e-09 8.969839e-09
 [6,]  2.9520356 -1.35475175  0.1899027 -2.8211220  0.12385813 -8.563461e-09 8.710930e-09
 [7,]  3.4689928 -0.76411068 -0.3016531  1.6369166 -1.94209512  7.151797e-09 2.945788e-09
 [8,]  0.3545235 -2.31408566 -2.2161772  2.9240116  2.00450379  1.388221e-09 5.158801e-09
 [9,] -2.9362323  0.01279597 -4.3117385 -2.5122743 -0.18911558  9.653831e-09 1.431683e-09
[10,]  1.9256952 -0.32526941  1.8734445 -1.6188611  0.90299062  6.240461e-09 5.703455e-09

$eig
 [1]  7.518716e+01  5.880560e+01  4.960516e+01  3.042789e+01  1.037419e+01  8.392572e-16
 [7]  3.002769e-16 -3.033284e-15 -3.907562e-15 -9.871675e-15

$x
NULL

$ac
[1] 0

$GOF
[1] 1 1
```

Recobramos las distancias con 5 vectores:

Ejemplo 1 IV

```
max(abs(dist(X) - dist(cmdscale(D,k=5))))
```

```
[1] 1.24345e-14
```

Para verificar en este ejemplo que obtenemos las mismas soluciones en PC que en MDS métrico (salvo signos, por eso tomamos valores absolutos):

```
max(abs(prcomp(X)$x) - abs(cmdscale(D,k=5)))
```

```
[1] 3.49807e-14
```

□

Excepciones a la solución métrica I

- Cuando la matriz de similitudes no es euclidiana, entonces la matriz **B** que se obtiene con la descomposición indicada **no es definida positiva**. En este caso:
  - algunos valores propios de **B** pueden ser negativos, lo que puede llevar a coordenadas complejas. En este caso, si **B** tiene pocos valores propios negativos, entonces se pueden usar los k más grandes positivos.
  - Como medida del ajuste se puede utilizar alguno de los siguientes criterios sugeridos por Mardia, Kent y Bibby (1979):
    - $P_k^{(1)} = \frac{\sum_{i=1}^{k} |\lambda_i|}{\sum_{i=1}^{n} |\lambda_i|}$, o
    - $P_k^{(2)} = \frac{\sum_{i=1}^{k} \lambda_i^2}{\sum_{i=1}^{n} \lambda_i^2}$
  - Un ajuste de 0.8 en este caso también se considera un buen ajuste.
- Cuando muchos de los valores propios de **B** son negativos, entonces se sugiere utilizar escalamiento multidimensional no-métrico.

Ejemplo 2 (no euclidiano) I

Ejemplo. [C]

onsideremos la misma matriz **X** pero ahora calculemos las distancias de Manhattan:

Ejemplo 2 (no euclidiano) II

```
Xm <- cmdscale(dist(X, method = "manhattan"), k = nrow(X)-1, eig = T)
```

```
Warning in cmdscale(dist(X, method = "manhattan"), k = nrow(X) - 1, eig = T): only 6 of the first 9 eigenvalues are > 0
```

```
Xm
```

```
$points
            [,1]       [,2]      [,3]         [,4]        [,5]        [,6]
 [1,]  5.8278669  1.5342565 -5.284199 -1.313859444 -0.44603914  1.83902117
 [2,] -4.0127510  9.7558670 -3.474399  0.001393465 -0.05363249 -0.69469808
 [3,] -7.4767187  4.1494313  6.876494 -0.171436000  0.71290321  2.50159974
 [4,] -8.1752216 -2.6283055 -1.568545  1.975015287 -0.20521998 -2.64967415
 [5,]  7.9887438  5.8314952 -1.160921  1.471164146 -1.10058263 -1.16547772
 [6,] -0.6454943 -6.2576861 -1.926460 -5.818551593 -0.47089688 -0.20612343
 [7,] -1.8889481 -5.4739596 -2.203709  3.311040551 -4.01247117  1.33827641
 [8,]  4.8998513 -4.3901118  3.820783  5.217725445  2.70513190  0.07147178
 [9,]  3.9779241 -0.2145464  9.619129 -3.265544068 -1.09258093 -1.25638659
[10,] -0.4952524 -2.3064406 -4.698175 -1.406947788  3.96338811  0.22199087

$eig
 [1]  2.806843e+02  2.494246e+02  2.288549e+02  9.250710e+01  4.250504e+01  2.196808e+01
 [7] -7.105427e-15 -1.507023e+01 -2.804630e+01 -5.682752e+01

$x
NULL

$ac
[1] 0

$GOF
[1] 0.901619 1.000000
```

Ejemplo 2 (no euclidiano) III

Calculando los criterios de bondad de ajuste $P_k^{(1)}$ y $P_k^{(2)}$, uno sugiere una solución en 4 dimensiones y la otra en tres:

```
Pk1 <- cumsum(abs(Xm$eig))/sum(abs(Xm$eig)); Pk1

 [1] 0.2762945 0.5218182 0.7470939 0.8381542 0.8799945 0.9016190 0.9016190 0.9164536 0.9440612
[10] 1.0000000

Pk2 <- cumsum(Xm$eig^2)/sum(Xm$eig^2); Pk2

 [1] 0.3779304 0.6763685 0.9276127 0.9686639 0.9773307 0.9796457 0.9796457 0.9807352 0.9845085
[10] 1.0000000
```

□

Ejemplo 3 (cráneos egipcios) I

- Los siguientes datos corresponden a 4 medidas de cráneos de hombres egipcios de 5 épocas diferentes. Se utilizarán estas medidas para 'mapear' los cráneos.
- Lo que interesa saber es si las medidas han cambiado con el tiempo. Esto podría indicar si hay cambios que hubo mezclas con las poblaciones inmigrantes.
- las medidas corresponden a las siguientes variables del cráneo: `mb`: ancho máximo, `bh`: medida basibregmática; `bl`: longitud basialiveolar y `nh`: altura nasal.

```
data("skulls", package="HSAUR")  #Paquete de Handbook of Stat Analysis Using R"
head(skulls)
```

```
    epoch  mb  bh  bl nh
1 c4000BC 131 138  89 49
2 c4000BC 125 131  92 48
3 c4000BC 131 132  99 50
4 c4000BC 119 132  96 44
5 c4000BC 136 143 100 54
6 c4000BC 138 137  89 56
```

- Para este ejemplo, se calcularán distancias de Mahalanobis entre cada par de épocas. se utilizará como matriz de covarianzas el estimador de la covarianza agrupada usual:

$$\mathbf{S} = \frac{29(\sum_{i=1}^{5} \mathbf{S}_i)}{149}$$

Ejemplo 3 (cráneos egipcios) II

```
# Calcula la varianza agrupada:
Svars <- tapply(1:nrow(skulls),skulls$epoch, function(i) var(skulls[i,-1]))
Spool <- 0
for (v in Svars) Spool <- Spool + 29*v
Spool <- Spool/149; Spool
```

```
           mb          bh          bl       nh
mb 20.54407159  0.03579418  0.07695749 1.955034
bh  0.03579418 22.85413870  5.06040268 2.768680
bl  0.07695749  5.06040268 23.52997763 1.102908
nh  1.95503356  2.76868009  1.10290828 9.880089
```

```
# Calcula las medias:

medias <- tapply(1:nrow(skulls), skulls$epoch,
                                 function(i)apply(skulls[i,-1],2,mean))
medias <- matrix(unlist(medias), nrow= length(medias), byrow=T)

# Calcula las distancias de Mahalanobis
DMahalanobis <- apply(medias,1, function(x)mahalanobis(medias,x,Spool))
DMahalanobis
```

```
          [,1]       [,2]      [,3]      [,4]      [,5]
[1,] 0.00000000 0.09354553 0.9279862 1.9330193 2.7712116
[2,] 0.09354553 0.00000000 0.7490463 1.6379863 2.2357084
[3,] 0.92798621 0.74904633 0.0000000 0.4553368 0.9359991
[4,] 1.93301928 1.63798631 0.4553368 0.0000000 0.2253343
[5,] 2.77121156 2.23570836 0.9359991 0.2253343 0.0000000
```

Ejemplo 3 (cráneos egipcios) III

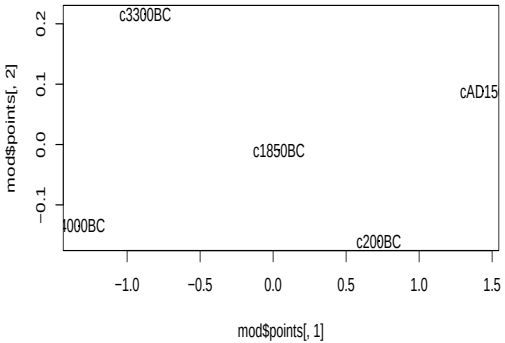
- Aplicando MDS a esta matriz de distancias, se tiene:

```
mod <- cmdscale(DMahalanobis, k=2, eig=T)
mod
```

```
$points
           [,1]         [,2]
[1,] -1.32692955 -0.134145042
[2,] -0.87427188  0.215462512
[3,]  0.04073376 -0.009300239
[4,]  0.72499547 -0.160637724
[5,]  1.43547219  0.088620493

$eig
[1]  5.112951e+00  9.816355e-02 -7.771561e-16 -1.008636e-01 -7.777015e-01

$x
NULL

$ac
[1] 0

$GOF
[1] 0.8557288 1.0000000
```

```
plot(mod$points[,1],mod$points[,2],pch=".")
text(mod$points[,1],mod$points[,2],labels=unique(skulls$epoch))
```

Ejemplo 3 (cráneos egipcios) IV



- La solución propuesta muestra el ordenamiento temporal de los cráneos, y es básicamente una solución unidimensional. Parece que hay un cambio en la forma de los cráneos a lo largo del tiempo. ¿Cuáles variables han cambiado más o menos?

Solución no métrica

Introducción

- Usualmente se basan en juicios ordinales (no de magnitudes o de distancias).
- Por ejemplo, ¿cómo se pueden percibir colores en la visión humana?: (Eckman, 1954): 14 colores que sólo difieren en matiz (ondas de 434 μm a 674 μm) se proyectan dos a la vez en un diseño de todos los posibles pares a 31 sujetos que calificaron cada uno de los posibles $\binom{14}{2} = 91$ pares, en una escala de 0 (no se parecen en nada) a 4 (identicos). La calificación para cada par se promedió sobre todos los sujetos y se escalaron al [0,1].
- Shepard (1962) y Kruskal (1964) desarrollaron un método de escalamiento multidimensional basado solo en rangos de las proximidades para producir una representación espacial.
- Este método es más complicado que la solución métrica, y es computacionalmente más elaborado. Pero también es más general, y permite resolver muchos problemas fácilmente.

Solución no métrica I

- Se tienen N items y hay $m = \binom{N}{2}$ similitudes entre pares de items. Sea $\mathcal{A}$ el conjunto de los N objetos y se tiene la similitud c_{ij} entre el i-ésimo y j-ésimo items.
- La idea es que las coordenadas en la representación espacial de las similitudes observadas da origen a unas distancias ajustadas d_{ij}, y que estas distancias están relacionadas a un conjunto de números que se llaman *disparidades*, $\hat{d}_{ij}$ a través de la relación:

$$d_{ij} = \hat{d}_{ij} + \epsilon_{ij}$$

donde las ϵ_{ij} son términos de error que incluyen medición y distorsión por la reducción de dimensión (de p a k).

- Se supone que las disparidades son monótonas con las similitudes observadas y entonces se parecen a las distancias ajustadas lo más posible: si

$$c_{i_1 j_1} < \quad ... \quad < c_{i_k j_k} \quad \text{o} \tag{1}$$
$$d_{i_1 j_1} > \quad ... \quad > d_{i_k j_k} \tag{2}$$

entonces

$$\hat{d}_{i_1 j_1} \leq \hat{d}_{i_2 j_2} \leq \cdots \leq \hat{d}_{i_m j_m}$$

Solución no métrica II

- Para encontrar las disparidades se utiliza regresión monotónica buscando minimizar un criterio basado en una función de pérdida (*stress* o *sstress*), que es función de la representación espacial de las similitudes observadas $\hat{\mathbf{X}}_{n\times k}$.
- Para cada dimensión k se puede obtener el stress mínimo: con una configuración de prueba en $\mathbb{R}^k$, se calculan los valores d_{ij}^k y $\hat{d}_{ij}^{(k)}$.

Bondad de ajuste I

- Usando $\hat{d}_{ij}$ para puntos en $\mathbb{R}^k$ se mueven los puntos para obtener una mejor configuración por un procedimiento de minimización aplicada a la función de pérdida S_k. Se espera que una nueva configuración tendrá nuevos valores $d's$ y menor *stress*. Los criterios propuestos por Kruskal pata evaluar el *stress* son los siguientes:

S_k	Ajuste
20 %	Pobre
10 %	Débil
5 %	Bueno
2.5 %	Excelente
0 %	Perfecto

En el caso de la función $SS_k \in [0, 1]$, se busca que tenga valores menores a 0.1.

Ejemplo1: votaciones I

- Los datos corresponden al número de veces que 15 congresistas de New Jersey votaron diferente sobre 19 leyes ambientales (Romesburg, 1984). No se registran abstenciones, pero dos congresistas se abstuvieron más frecuentemente que el resto: Sandman (9) y Thompson (6).

```
data("voting", package = "HSAUR2")
head(voting)
```

```
                  Hunt(R) Sandman(R) Howard(D) Thompson(D) Freylinghuysen(R) Forsythe(R) Widnall(R)
Hunt(R)                 0          8        15          15                10          9          7
Sandman(R)              8          0        17          12                13         13         12
Howard(D)              15         17         0           9                16         12         15
Thompson(D)            15         12         9           0                14         12         13
Freylinghuysen(R)      10         13        16          14                 0          8          9
Forsythe(R)             9         13        12          12                 8          0          7
                  Roe(D) Heltoski(D) Rodino(D) Minish(D) Rinaldo(R) Maraziti(R) Daniels(D)
Hunt(R)               15          16        14        15         16           7         11
Sandman(R)            16          17        15        16         17          13         12
Howard(D)              5           5         6         5          4          11         10
Thompson(D)           10           8         8         8          6          15         10
Freylinghuysen(R)     13          14        12        12         12          10         11
Forsythe(R)           12          11        10         9         10           6          6
                  Patten(D)
Hunt(R)              13
Sandman(R)           16
Howard(D)             7
Thompson(D)           7
Freylinghuysen(R)    11
Forsythe(R)          10
```

Ejemplo1: votaciones II

- El objetivo de interés es si se pueden detectar afiliaciones a los partidos en los datos.
- La función para estimar MDS no métrico es `isoMDS` en el paquete `MASS`

```
library(MASS)
votacion <- isoMDS(voting)
```

```
initial  value 15.268246
iter   5 value 10.264075
final  value 9.879047
converged
```

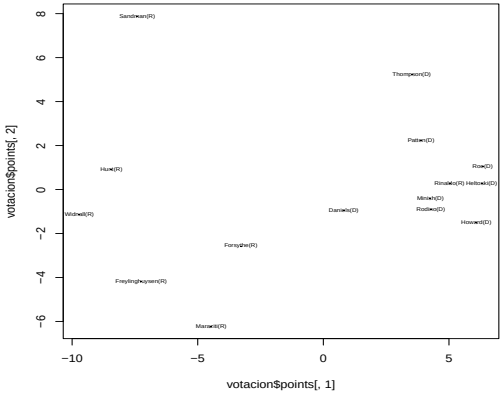
```
votacion
```

```
$points
                          [,1]       [,2]
Hunt(R)            -8.4354008  0.9063380
Sandman(R)         -7.4050250  7.8770232
Howard(D)           6.0930164 -1.4971986
Thompson(D)         3.5187022  5.2486888
Freylinghuysen(R) -7.2457425 -4.1821704
Forsythe(R)        -3.2787096 -2.5689673
Widnall(R)         -9.7110008 -1.1187710
Roe(D)              6.3429759  1.0388694
Heltoski(D)         6.2983842  0.2706499
Rodino(D)           4.2829160 -0.9151604
Minish(D)           4.2642545 -0.3919690
Rinaldo(R)          5.0285425  0.2665701
Maraziti(R)        -4.4577693 -6.2177727
Daniels(D)          0.8129854 -0.9417672
Patten(D)           3.8918709  2.2256372

$stress
[1] 9.879047
```

Ejemplo1: votaciones III

- Graficamos la solución para dos dimensiones:

```
plot(votacion$points[,1],votacion$points[,2],pch=".")
text(votacion$points[,1],votacion$points[,2],labels=colnames(voting),cex=0.5)
```



Ejemplo1: votaciones IV

- En los datos se puede ver que el republicano Rinaldo está más en línea con los demócratas que con otros miembros de su partido.
- En general se puede ver que los republicanos tienen mayor variación que los demócratas.

Ejemplo2: Opiniones sobre líderes en la WWII I

- En este conjunto de datos se pretende obtener una representación espacial de las opiniones sobre las disimilitudes ideológicas de los líderes y políticos prominentes en la WWII. La evaluación se realiza sobre una escala de 9 puntos, donde 1 es 'muy similar' y 9 es 'muy diferente'

```
WWIIlideres <- c(3, 4, 6, 7, 8, 4, 3, 5, 6, 8, 8, 9, 3, 9, 8,  3, 2, 5, 7, 6, 7,  4, 4, 3, 5, 6, 5, 4,
                           8, 9, 8, 9, 6, 9, 8, 7, 9, 9, 5, 4, 7, 8, 8, 4, 4, 4, 5, 5, 4, 7, 2, 2, 5, 9, 5,
                           7, 8, 2, 4, 7, 8, 3, 2, 4, 5, 7)
tmp <- matrix(0, ncol = 12, nrow = 12)
tmp[upper.tri(tmp)] <- WWIIlideres
tmp <- tmp + t(tmp)
rownames(tmp) <- colnames(tmp) <- c("Hitler", "Mussolini", "Churchill", "Eisenhower", "Stalin", "Attlee",
                                    "Franco", "De Gaulle", "Mao Tse-Tung","Truman", "Chamberlin", "Tito")
WWIIlideres <- as.dist(tmp)
```

- La solución no métrica:

Ejemplo2: Opiniones sobre líderes en la WWII II

```
WWII <- isoMDS(WWIIlideres)
```

```
initial  value 20.504211
iter   5 value 15.216103
iter   5 value 15.207237
iter   5 value 15.207237
final  value 15.207237
converged
```

```
WWII
```

```
$points
                  [,1]        [,2]
Hitler       -2.5820321 -1.75960714
Mussolini    -3.8806920 -1.24755866
Churchill     0.3109807  1.46671155
Eisenhower    2.9852347  2.87821624
Stalin       -1.4273957 -3.75699052
Attlee       -2.1067050  5.07317056
Franco       -2.8589747  0.07877559
De Gaulle     0.6590874 -0.20655989
Mao Tse-Tung  4.1604539 -4.57583381
Truman        4.4961515  0.29294331
Chamberlin   -2.1419835  2.75876703
Tito          2.3858746 -1.00203426

$stress
[1] 15.20724
```

Ejemplo2: Opiniones sobre líderes en la WWII III

```
x <- WWII$points[,1]
y <- WWII$points[,2]
par(pty="s")
plot(x, y, pch = ".")
text(x, y, labels = labels(WWIIlideres), cex = 0.7)
abline(h=0,v=0)
```

Ejemplo2: Opiniones sobre líderes en la WWII IV