Clustering com

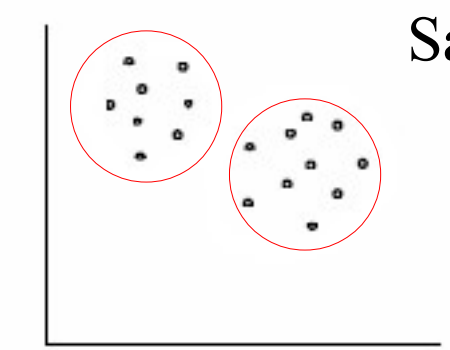
Bioinformática 2019/2020
Marta Mesquita

Introdução

- Dado um conjunto de n indivíduos pretende-se identificar grupos (ou classes) de individuos que reflictam características presentes num conjunto de dados.
- Não se dispõe de qualquer informação a-priori sobre estes grupos.
- Indivíduos na mesma classe devem ter características semelhantes. Indivíduos em classes diferentes deverão ser dissemelhantes.
- Homogeneidade: cada classe deve ser internamente homogéneo
- Separação: classes diferentes devem ser heterogéneas entre si
- Uma lista de packages e funções do R que podem ser utilizados em análise classificatória está disponível em

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

Introdução



Satisfaz homogeneidade e separação

Não satisfaz homogeneidade e separação

Medidas de semelhança (dissemelhança)

- Numa análise classificatória pretende-se identificar indivíduos que sejam semelhantes e/ou indivíduos que sejam dissemelhantes.
- O grau de semelhança (dissemelhança) entre indivíduos vai depender da medida que é escolhida para avaliar essa semelhança (dissemelhança).
- O conhecimento do problema é útil para a escolha da medida que vai ser utilizada.
- Uma medida de dissemelhança d_{ij} entre o indivíduo *i* e o indivíduo *j* satisfaz as seguintes propriedades:

$$d_{i,j} \geqslant 0; \quad d_{i,i} = 0; \quad d_{i,j} = d_{j,i}$$

- Uma medida de distância satisfaz

$$d_{i,j} \geqslant 0; \quad d_{i,i} = 0; \quad d_{i,j} = d_{j,i}; \quad d_{i,j} \leqslant d_{i,k} + d_{k,j}$$

Exemplos de distâncias

- Seja d_{ij} a distância entre os indivíduos i e j, construída a partir de valores de p variáveis, traduzidos nos vectores $x(i)$ e $x(j)$.
- Métrica euclideana $$d_{i,j} = \sqrt{\sum_{k=1}^{p} (x_{ik} - x_{jk})^2}$$

```
> ?dist

> dist(x, method = "euclidean", diag = FALSE,
+ upper = FALSE, p = 2)

# method - the distance measure to be used.

# This must be one of "euclidean", "maximum",
# "manhattan", "canberra", "binary" or

# "minkowski".
```

Exemplos de distâncias

- Métrica de Minkowski $d_{i,j}=\left(\sum_{k=1}^{p}|x_{ik}-x_{jk}|^{\lambda}\right)^{\frac{1}{\lambda}}$

  tem como casos particulares a métrica euclideana $(\lambda=2)$, de Manhattan $(\lambda=1)$ e do máximo $(\lambda=\infty)$

- Métrica de Manhattan ou city-block $d_{i,j}=\sum_{k=1}^{p}|x_{ik}-x_{jk}|$

- Métrica do máximo $d_{i,j}=\max_{k}|x_{ik}-x_{jk}|$

- Métrica de Canberra (variáveis com valores não negativos) $d_{i,j}=\sum_{k=1}^{p}\frac{|x_{ik}-x_{jk}|}{(x_{ik}+x_{jk})}$

  tem por objectivo obter uma medida invariante a transformações de escalas

Medida de (dis)semelhança para dados binários

- No caso de dados binários (apenas tomam dois valores diferentes: 1/0 ou Sim/Não ou ...) o R tem disponível a distância correspondente à medida de semelhança dada pelo Coeficiente de Jacard

$$s_{i,j} = \frac{a}{a+b+c}$$

em que

a – número de variáveis em que ambos os indivíduos tomam o valor 1

b – número de variáveis em que *i* toma valor 1 e *j* toma valor 0

c – número de variáveis em que *i* toma valor 0 e *j* toma valor

- No R tem-se

$$d_{i,j} = 1 - \frac{a}{a+b+c} = \frac{b+c}{a+b+c}$$

(número de variáveis em que apenas 1 dos indivíduos tem valor 1 a dividir pelo número de variáveis em que pelo menos um dos indivíduos tem valor 1)

exemplos

```
x <- matrix(rnorm(10), nrow=5)
x
dist(x)
dist(x, diag = T, upper=T)  ## euclideana
dist(x, method = "maximum", diag = T, upper=T)

## examplo para dados binários.
x <- c(0, 0, 1, 1, 1, 1)
y <- c(1, 0, 1, 1, 0, 1)
dist(rbind(x,y), method= "binary")
##  0.4 = 2/5
```

Métodos de classificação

- Podemos considerar que existem três tipos de métodos de classificação:

  - Métodos hierárquicos – criam uma hierarquia de conjuntos de classes em cujos extremos temos a classificação de cada indivíduo como uma classe e a classificação de todos os indivíduos na mesma classe.

  - Métodos não hierárquicos de partição – determinam uma partição dos indivíduos em k classes que optimize um critério de homogeneidade e/ou separação das classes. Em geral, necessitam como "input" do número k de classes que se vai criar.

  - Métodos "fuzzy" - não se limita o número de classes a que um indivíduo pode pertencer. O indivíduo *i* pertence à classes C_j com probabilidade p_{ij}.

Métodos hierárquicos

- Criam uma hierarquia de conjuntos de classes por fusão de classes mais pequenas em classes maiores (ascendente) ou por divisão de classes maiores em classes mais pequenas (descendente).
- O resultado de um algoritmo hierárquico é uma árvore ou dendograma.
- Cortando a árvore num determinado nível obtem-se uma partição dos indivíduos em k classes.
- **Hierárquicos aglomerativos**: Partem de n individuos agrupados em n classes, cada classe com 1 indivíduo. Agrupam as classes sucessivamente até se obter uma única classe.
- **Hierárquicos divisivos**: Partem de uma única classe que inclui os n indivíduos. As classes são sucessivamente divididas em classes "mais pequenas" até se obterem n classes, cada uma com um indivíduo.

Distância entre classes

- Vizinho mais próximo (nearest neighbour ou single-linkage)

  a distância entre as classes G e H é dada pelo mínimo das distâncias entre um elemento da classe G e um elemento da classe H

$$D_{GH} = \min_{i \in G,\, j \in H} d_{ij}$$

  Tende a produzir classes alongadas, com indivíduos que podem estar "muito" distantes entre si

- Vizinho mais distante (furthest neighbour ou complete-linkage)

  a distância entre as classes G e H é dada pelo máximo das distâncias entre um elemento da classe G e um elemento da classe H

$$D_{GH} = \max_{i \in G,\, j \in H} d_{ij}$$

  Tende a produzir classes "esféricas"

Distância entre classes

- Distâncias médias entre classes G e H (group average ou average linkage, UPGMA) é a média de todas as distâncias entre pares de elementos, um de G e um de H

$$D_{GH} = \frac{1}{n_G n_H} \sum_{i=1}^{n_G} \sum_{j=1}^{n_H} d_{ij}$$

  Tende a produzir classes "esféricas"

- Método de Ward (método da inércia mínima) . Baseado no conceito de inércia de uma classe. A inércia de uma classe G é dada pela soma dos quadrados das diferenças entre cada indivíduo e o centro de gravidade "da classe" (indivíduo "médio" da classe). A fusão de classes tem por objectivo minimizar a variância intra-classes.

  Tende a produzir classes com um número aproximadamente igual de indivíduos.

Distância entre classes

- Método dos centróides. A distância entre duas classes é dada pela distância entre os respectivos centros de gravidade ou outros pontos representativos das classes (centróides).

- A escolha das medidas de "distância" a utilizar para agrupar indivíduos e classes vai condicionar a solução final. Esta escolha deve ter em conta a natureza dos dados e o objectivo da classificação. Será útil experimentar diferentes medidas a fim de verificar se há robustez na classificação final.

Método hierárquico aglomerativo no R

```
> y <-  iris[6*(1:25),-5]

## sub data.frame de iris

> z <- iris$Species[6*(1:25)]

> table(z)  ## conta numero de ocorrencias de cada
valor


> ?hclust

> hr<-hclust(dist(y),method="single"); hr

> str(hr)

> hr$merge  ##  distancias entre clusters

> hr$height ## descreve a fusão
```

Método hierárquico aglomerativo no R

- $merge

  an n-1 by 2 matrix. Row i of merge describes the merging of clusters at step i of the clustering. If an element j in the row is negative, then observation -j was merged at this stage. If j is positive then the merge was with the cluster formed at the (earlier) stage j of the algorithm. Thus negative entries in merge indicate agglomerations of singletons, and positive entries indicate agglomerations of non-singletons.

- $height

  a set of n-1 real values (non-decreasing for ultrametric trees). The clustering height: that is, the value of the criterion associated with the clustering method for the particular agglomeration.

Método hierárquico aglomerativo no R

```
> par(mfrow=c(1,2))
> plot(hclust(dist(y),method="single"))
> plot(hclust(dist(y),method="single"),hang=-1)
## folhas ficam tds ao mesmo nivel


> plot(as.dendrogram(hclust(dist(y),method="single"))
 + ,horiz=T, main="dist(y)\nSingle")
 ## visualização horizontal
```

Método hierárquico aglomerativo no R

```
## visualização de dendogramas construídos com
## diferentes "métodos"

> par(mfrow=c(2,3))

> plot(hclust(dist(y),method="single"))

> plot(hclust(dist(y),method="complete"))

> plot(hclust(dist(y),method="average"))


> plot(hclust(dist(y,method="maximum"),method="single"))

> plot(hclust(dist(y,method="maximum"),method="complete"))

> plot(hclust(dist(y,method="maximum"),method="average"))
```

- Num método aglomerativo, um par de indivíduos que, numa etapa, seja incluído na mesma classe não poderá ser separado em etapas posteriores.

Método hierárquico aglomerativo no R

```
  ## visualização das classes
> hclust.avg <- hclust(dist(y),method="average")
> plot(hclust.avg)
> rect.hclust(hclust.avg, k=3, border="red")
> classes<-cutree(hclust.avg,3) ## considera 3 classes
> classes
> table(classes)  ### diz numero de elementos por classe
> table(rownames(y),classe=classes)
```

- Uma “boa” classificação poderá ser obtida por cortar o dendograma numa zona onde as separações entre classes correspondam a grandes distâncias .

Método hierárquico divisivo no R

```
> library(cluster)

> ?diana
```

The diana-algorithm constructs a hierarchy of clusterings, starting with one large cluster containing all n observations. Clusters are divided until each cluster contains only a single observation.

At each stage, the cluster with the largest diameter is selected. (The diameter of a cluster is the largest dissimilarity between any two of its observations.)

To divide the selected cluster, the algorithm first looks for its most disparate observation (i.e., which has the largest average dissimilarity to the other observations of the selected cluster). This observation initiates the "splinter group". In subsequent steps, the algorithm reassigns observations that are closer to the "splinter group" than to the "old party". The result is a division of the selected cluster into two new clusters.

Método hierárquico divisivo no R

```
> diana.result <- diana(dist(y))
> diana.result
> plot(diana.result)
> methods(plot) ## para ver plot.diana
> ?plot.diana
 ## para ver apenas o dendograma
> plot(diana.result,which.plots = 2)
```

Árvores filogenéticas

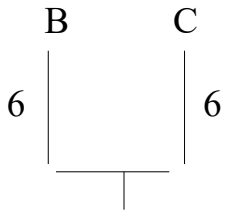
- Os métodos para a construcção de árvores filogenéticas dividem-se de acordo com o tipo de dados e o tipo de método utilizado na construcção da árvore.
- No caso em que os dados de partida são distâncias os algoritmos de clustering mais utilizados na construção de árvores filogenéticas são **UPGMA** (unweighted pair-group method with arithmetic mean) e **NJ** (neighbor joining).
- UPGMA é adequado para construir uma árvore de linhagens com taxas de evolução constantes
- NJ não assume que todas as linhagens evoluem à mesma taxa

Árvores filogenéticas - UPGMA

- Exemplo:

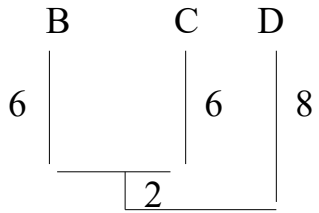
16/2=8; 8+6+2=16

	A	B	C	D
A	0			
B	17	0		
C	21	12	0	
D	27	18	14	0



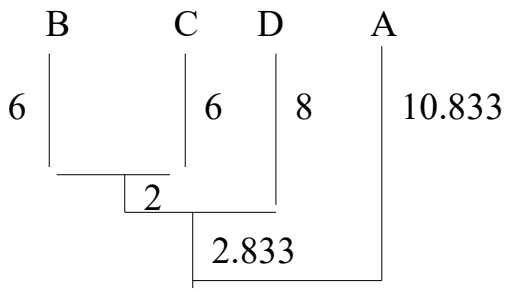
$$min\, d_{ij} = 12, \quad i \neq j$$

	A	BC	D
A	0		
BC	19	0	
D	27	16	0



$$d_{BC,A} = \frac{d_{B,A} + d_{C,A}}{2} = \frac{17+21}{2}$$

	A	BCD
A	0	
BCD	21.7	0



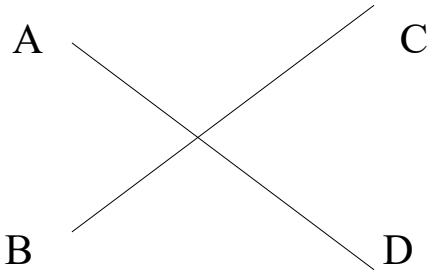
$$d_{BCD,A} = \frac{d_{B,A} + d_{C,A} + d_{D,A}}{3} = \frac{17+21+27}{3} = 21.66667$$

21.666667/2=10.833

Árvores filogenéticas – NJ

	A	B	C	D
A	0			
B	17	0		
C	21	12	0	
D	27	18	14	0

ui
32.5
23.5
23.5
29.5



PASSO 1. Para cada classe i calcular $u_i = \sum_{j=1:\, j \neq i}^{n} \frac{D_{ij}}{(n-2)}$ $u_A = \frac{17+21+27}{2} = 32.5$

Medida da separação da classe i às outras classes

PASSO 2. Para cada par (i,j) calcular $D_{ij} - u_i - u_j$ e escolher par (i,j) correspondente ao menor valor

	A	B	C	D
A				
B	-39			
C	-35	-35		
D	-35	-35	-39	

juntar (A,B) e (C,D)

Árvores filogenéticas – NJ

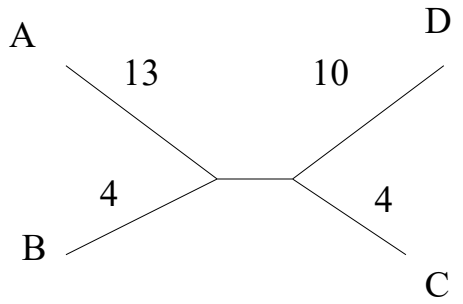
Juntar A e B numa classe e C e D noutra classe

PASSO 3. Calcular o comprimento do ramo i ao novo nó (i,j) de acordo com as fórmulas:

$$V_{(i,ij)}=\frac{1}{2}D_{ij}+\frac{1}{2}(u_i-u_j) \qquad V_{(j,ij)}=\frac{1}{2}D_{ij}+\frac{1}{2}(u_j-u_i)$$

$$V_{(A,AB)}=\frac{1}{2}\times 17+\frac{1}{2}\times(32.5-23.5)=13$$

$$V_{(B,AB)}=4;\quad V_{(B,AB)}=4; V_{(C,CD)}=4; V_{(D,CD)}=10;$$



Árvores filogenéticas – NJ

PASSO 4. Calculara distância entre o novo nó (i,j) e os restantes nós k:

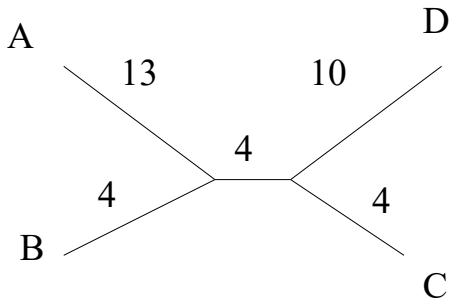
$$D_{(ij),k} = \frac{D_{ik} + D_{jk} - D_{ij}}{2}$$

$$D_{(AB,C)} = \frac{21 + 12 - 17}{2} = 8$$

$$D_{(AB,D)} = \frac{27 + 18 - 17}{2} = 14$$

$$D_{(CD,A)} = \frac{21 + 27 - 14}{2} = 17$$

$$D_{(CD,B)} = \frac{12 + 18 - 14}{2} = 8$$



Árvores filogenéticas – UPGMA, NJ

- Fazer exemplos em:

http://evolution-textbook.org/content/free/tables/Ch_27/T9_EVOW_Ch27.jpg

http://evolution-textbook.org/content/free/tables/Ch_27/T11_EVOW_Ch27.jpg

Métodos não hierárquicos de partição

- Os métodos não hierárquicos de partição têm como objectivo obter uma partição dos dados num conjunto de classes que optimize algum critério de homogeneidade interna e heterogeneidade externa.
- Geralmente, é necessário definir à partida o número de classes
- Os métodos de partição partem de um conjunto de k pontos (representantes das classes ou centróides). Em geral, o método itera entre a afectação dos indivíduos aos centróides (Passo 1) e o re-cálculo de novos centróides (Passo 2) até que um determinado critério de paragem seja verdadeiro.

  Passo 1: Calcula a distância dos indivíduos aos centróides (representantes de cada classe). (Re-)afecta cada indivíduo ao centróide "mais semelhante";

  Passo 2: (Re-)calcula o centróide de cada classe;

  Critério de paragem: não existem alterações no conjunto de classes em duas iterações sucessivas.

Método não hierárquico de partição kmeans

- K-means (Mc Queen, 1967) – Necessita do número de classes como parâmetro de entrada. Pretende-se minimizar a soma dos quadrados das distâncias de cada indivíduo ao centróide da classe a que pertence.

```
> ?kmeans

kmeans(x, centers, iter.max = 10, nstart = 1, algorithm
= c("Hartigan-Wong", "Lloyd", "Forgy","MacQueen"))

> kresult<-kmeans(y,3) ## número de classes = 3

> kresult$centers  ## coordenadas dos centros das classes

> kresult$cluster  ## mostra a classe de cada elemento

> table(rownames(y),kresult$cluster)

## número de elementos por classe
```

`kmeans` – representação gráfica

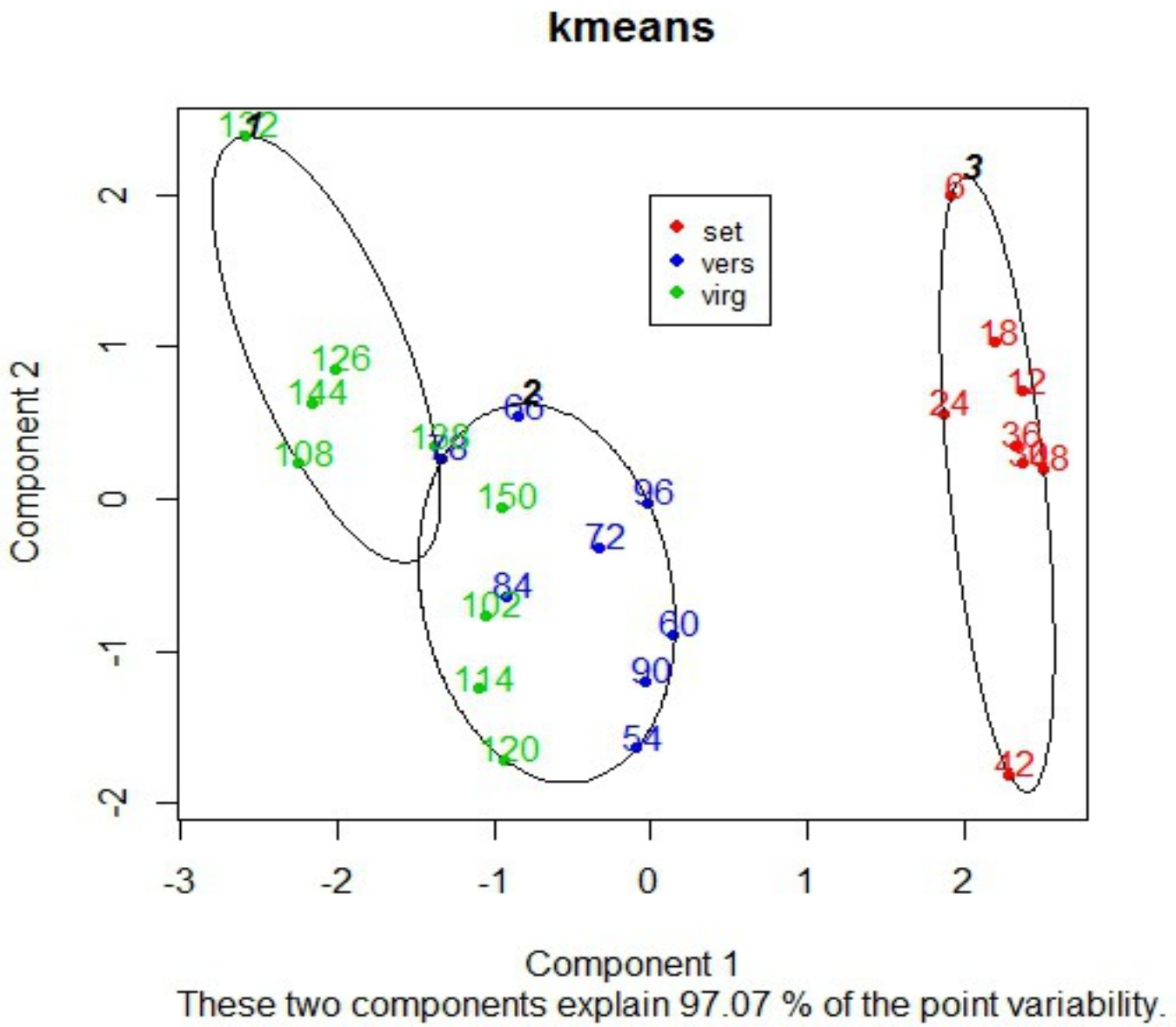
```
> clus<-kresult$cluster


> clusplot(y,clus)
> ?clusplot # ?clusplot.default
> clusplot(y,clus,col.p=clus,main="kmeans",
+ lines=0) ## sem linhas a ligar classes
> clusplot(y,clus,col.p=clus,main="kmeans",
+ lines=0,labels=4)    ## com o numero da
classe
```

`kmeans` – representação gráfica

```
> clusplot(y,clus,col.p=clus,main="kmeans",
+ lines=0,labels=2,col.clus=c(1,1,1))
# linhas das elipses a preto
> clusplot(y,clus,col.p=rep(c("red","blue",
+ "green3"),c(8,8,9)),main="kmeans",lines=0,
+ labels=2,col.clus=c(1,1,1),
+ plotchar=F,pch=16)
> legend(0,2,c("set","vers","virg"),col=
+ c("red","blue","green3"), pch =
+ c(16, 16, +16),cex=0.8)
```

`kmeans` – representação gráfica



Métodos não hierárquico de partição `PAM`

- PAM - Partioning Around Medoids (Kaufman & Rousseau, 1990) – Difere do método k-means no que respeita a escolha dos k representantes das classes e na função a minimizar. Os k representantes das classes são escolhidos entre os indivíduos observados.
- No R a função `pam()` aceita uma matriz de dissemelhanças e permite minimizar a soma de dissemelhanças associadas a métricas diferentes da euclideana.

```
> cluspam<-pam(y,3)

> summary(cluspam)

> str(cluspam)

> cluspam$medoids

> cluspam$id.med

> cluspam$clustering

> cluspam$clusinfo
```

PAM – representação gráfica

```
> par(mfrow=c(1,2))  ## comparar kmeans com pam
### kmeans
>clusplot(y,clus,col.p=rep(c("red","blue","green3"),
c(8,8,9)),main="kmeans",lines=0,plotchar=F,pch=16,
+ labels=2,col.clus=c(1,1,1),)


### pam
> clusp<-cluspam$clustering
> clusplot(y,clusp,col.p=rep(c("red","blue","green3"),
+ c(8,8,9)),main="pam",lines=0,plotchar=F,pch=16,
+ labels=2,col.clus=c(1,1,1),)
```

Escolha do número de classes

- Qual o número de classes k que deve ser considerado?
- Qual a "qualidade" dos agrupamentos?
- Rousseeuw (1987) sugere um método gráfico "Silhouette plot" que pode ser usado para responder a estas questões.
- Para cada observação i calcula-se:

  - Seja A a classe de i. Calcula-se a distância média entre i e os restantes indivíduos de A, a(i)

  - Para cada uma das classes C diferentes de A, calcula-se a distância média entre i e os indivíduos de C. Seja b(i) o mínimo destas distâncias.

  - Para o indivíduo i, o valor de "silhouette" é dado por

$$s(i) = \frac{b(i) - a(i)}{max(a(i), b(i))}$$

silhouette

- Para o indivíduo i, o valor de "silhouette" é dado por

$$s(i) = \frac{b(i)-a(i)}{max(a(i),b(i))} \qquad -1 \leq s(i) \leq 1$$

- Valores negativos de s(i) negativos sugerem que o indivíduo i seja semelhante a indivíduos de outras classes.
- Valores de s(i) perto de 1 sugerem que i esteja bem classificado.

silhouette

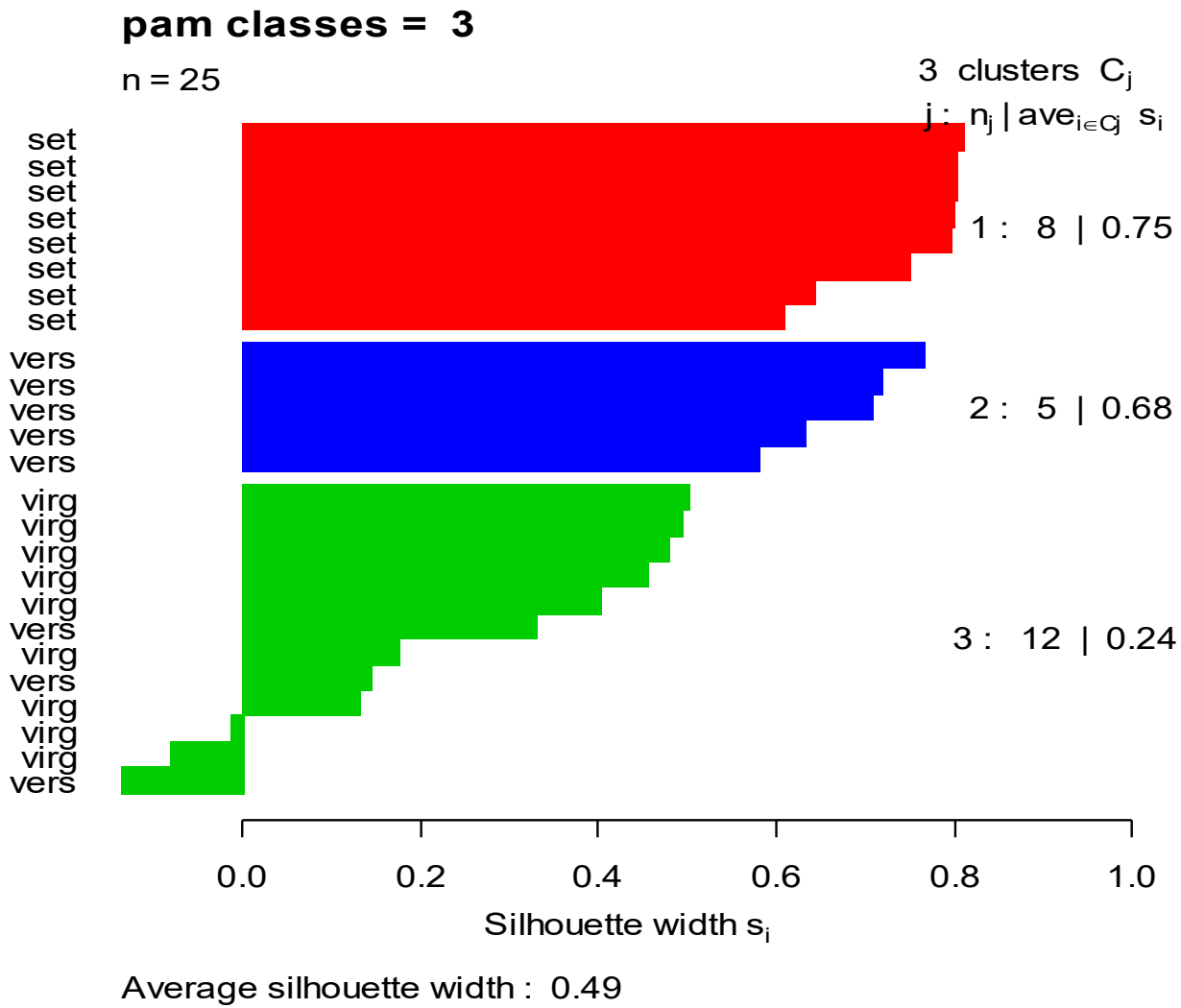
```
> cluspam<-pam(y,3)
> summary(cluspam)
Silhouette plot information:
     cluster neighbor   sil_width
set        1        2  0.80830402
set        1        2  0.80163202
…
Average silhouette width per cluster:
[1] 0.7509041 0.6795037 0.2396307
Average silhouette width of total data set:
[1] 0.4912128
```

silhouette – representação gráfica

```
> cluspam<-pam(y,3)
> summary(cluspam)
> si<-silhouette(cluspam)
> plot(si,col=1:3)


> par(mfrow=c(2,2)) ## escolha de classes
> a<-c()
> for (i in 2:4){ si<-silhouette(pam(y,i))
 plot(si,col=1:i,main=paste("classes = ",i))
 ssi<-summary(si)
 a<-c(a,ssi$avg.width) }
> plot(2:4,a)
```

`silhouette` – representação gráfica



silhouette – representação gráfica

```
#### hierárquico aglomerativo hclust
> par(mfrow=c(2,2))
> a<-hclust(dist(y))
> plot(a,hang=-1,main="aglomerativo")
> a1<-silhouette(cutree(a,k=3),dist(y))
> plot(a1,col=1:3,main="aglomerativo")

#### hierárquico divisivo diana
> a<-diana(dist(y))
> plot(a,which.plots = 2,hang=-1,main="divisivo")
> a1<-silhouette(cutree(a,k=3),dist(y))
> plot(a1,col=1:3,main="divisivo")
```

Bibliografia

- Material de apoio à disciplina de Estatística Multivariada, Capítulo 4 (texto), Mestrado em Matemática Aplidada às Ciências Biológicas, Jorge Cadima, 2009/2010.

  http://www.isa.utl.pt/dm/mestrado/2009-10/UCs/em/webEMCap4.pdf

- Package 'cluster', Version 2.0.7-1, Date 2018-04-06

  http://cran.r-project.org/web/packages/cluster/cluster.pdf