Ciencia de Datos

Victor Muñiz

victor_m@cimat.mx

Asistente:

Víctor Gómez

victor.gomez@cimat.mx

Maestría en Cómputo Estadístico.

Centro de Investigación en Matemáticas.

Unidad Monterrey.

Enero-Junio 2021

Clustering con algoritmos combinatorios

Clustering basado en algoritmos combinatorios

Ya que clustering no es un método supervisado, no podemos obtener directamente una medida de "error" al asignar clusters, sin embargo, podemos definir criterios deseables para optimizar.

En general, queremos:

- objetos **dentro** de un clúster sean parecidos
- los clústers estén separados, o equivalentemente, objetos **entre** clústers sean diferentes

Clustering basado en algoritmos combinatorios

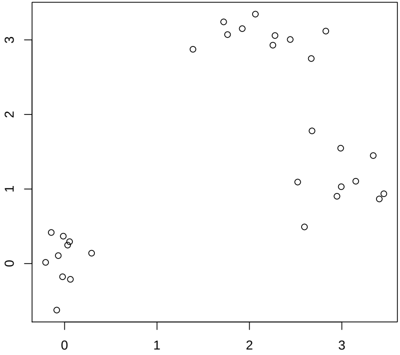
Ya que clustering no es un método supervisado, no podemos obtener directamente una medida de "error" al asignar clusters, sin embargo, podemos definir criterios deseables para optimizar.

En general, queremos:

- objetos **dentro** de un clúster sean parecidos
- los clústers estén separados, o equivalentemente, objetos **entre** clústers sean diferentes

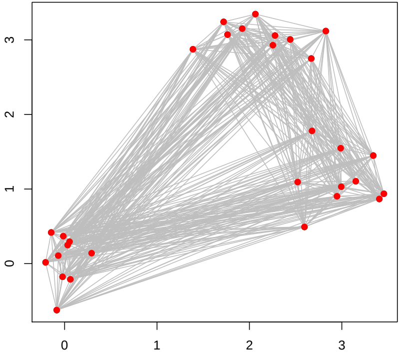
Clustering basado en algoritmos combinatorios

Estas características las cuantificamos usando lo que ya conocemos...



Clustering basado en algoritmos combinatorios

Estas características las cuantificamos usando lo que ya conocemos... **disimilaridades**



Clustering basado en algoritmos combinatorios

Definimos la disimilaridad total (total point scatter) como

$$T = \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} d_{ij}.$$

Observa que T es *constante dados los datos*.

Definimos también el encoder $C(i)$ que asigna la observación i a algún cluster k:

$$k = C(i),$$

$k \in \{1, 2, \ldots, K\}$, $i = 1, \ldots, n$

Clustering basado en algoritmos combinatorios

Definimos la disimilaridad total (total point scatter) como

$$T = \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} d_{ij}.$$

Observa que T es *constante dados los datos*.
Definimos también el encoder $C(i)$ que asigna la observación i a algún cluster k:

$$k = C(i),$$

$k \in \{1, 2, \ldots, K\}$, $i = 1, \ldots, n$

Clustering basado en algoritmos combinatorios

Entonces

$$
\begin{aligned}
T &= \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} d_{ij} \\
&= \frac{1}{2}\sum_{k=1}^{K}\sum_{C(i)=k}\left(\sum_{C(j)=k} d_{ij} + \sum_{C(j)\neq k} d_{ij}\right) \\
&= \frac{1}{2}\sum_{k=1}^{K}\sum_{C(i)=k}\underbrace{\sum_{C(j)=k} d_{ij}}_{\text{están en cluster } k} + \frac{1}{2}\sum_{k=1}^{K}\sum_{C(i)=k}\underbrace{\sum_{C(j)\neq k} d_{i}}_{\text{no están en clu}}
\end{aligned}
$$

Clustering basado en algoritmos combinatorios

Entonces

$$
\begin{aligned}
T &= \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} d_{ij} \\
&= \frac{1}{2}\sum_{k=1}^{K}\sum_{C(i)=k}\left(\sum_{C(j)=k} d_{ij} + \sum_{C(j)\neq k} d_{ij}\right) \\
&= \frac{1}{2}\sum_{k=1}^{K}\sum_{C(i)=k}\underbrace{\sum_{C(j)=k} d_{ij}}_{\text{están en cluster } k} + \frac{1}{2}\sum_{k=1}^{K}\sum_{C(i)=k}\underbrace{\sum_{C(j)\neq k} d_{ij}}_{\text{no están en clu}}
\end{aligned}
$$

Clustering basado en algoritmos combinatorios

Entonces

$$
\begin{aligned}
T &= \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} d_{ij} \\
&= \frac{1}{2}\sum_{k=1}^{K}\sum_{C(i)=k}\left(\sum_{C(j)=k} d_{ij} + \sum_{C(j)\neq k} d_{ij}\right) \\
&= \frac{1}{2}\sum_{k=1}^{K}\sum_{C(i)=k}\underbrace{\sum_{C(j)=k} d_{ij}}_{\text{están en cluster } k} + \frac{1}{2}\sum_{k=1}^{K}\sum_{C(i)=k}\underbrace{\sum_{C(j)\neq k} d_{i}}_{\text{no están en clu}}
\end{aligned}
$$

Clustering basado en algoritmos combinatorios

Por lo tanto,

$$\underbrace{T}_{\text{Disim. Total}} = \underbrace{W(C)}_{\text{Disim. Dentro}} + \underbrace{B(C)}_{\text{Disim. Entre}}$$

Queremos

$$\min W(C)$$
$$\max B(C).$$

Dados los datos $\mathbf{x}_i$, $i = 1, \ldots, n$, es fácil ver que

$$\min_C W(C) = T - B(C)$$

es equivalente a

$$\max_C B(C),$$

y viceversa.

Clustering basado en algoritmos combinatorios

Por lo tanto,

$$\underbrace{T}_{\text{Disim. Total}} = \underbrace{W(C)}_{\text{Disim. Dentro}} + \underbrace{B(C)}_{\text{Disim. Entre}}$$

Queremos

$$\min W(C)$$
$$\max B(C).$$

Dados los datos $\mathbf{x}_i$, $i = 1, \ldots, n$, es fácil ver que

$$\min_C W(C) = T - B(C)$$

es equivalente a

$$\max_C B(C),$$

y viceversa.

Clustering basado en algoritmos combinatorios

Por lo tanto,

$$\underbrace{T}_{\text{Disim. Total}} = \underbrace{W(C)}_{\text{Disim. Dentro}} + \underbrace{B(C)}_{\text{Disim. Entre}}$$

Queremos

$$\begin{gathered}\min W(C)\\ \max B(C).\end{gathered}$$

Dados los datos $\mathbf{x}_i$, $i = 1, \ldots, n$, es fácil ver que

$$\min_C W(C) = T - B(C)$$

es equivalente a

$$\max_C B(C),$$

y viceversa.

Clustering basado en algoritmos combinatorios

La solución del problema de optimización anterior usando algoritmos combinatorios puede ser computacionalmente muy compleja para datos relativamente grandes.

Los algoritmos prácticos de clustering examinan solo una parte de todas las posibles asignaciones $k = C(i)$, por lo tanto, convergen a óptimos locales.

Entre los más conocidos, tenemos

- $k-$means
- $k-$medoids
- fuzzy $k-$means

Clustering basado en algoritmos combinatorios

La solución del problema de optimización anterior usando algoritmos combinatorios puede ser computacionalmente muy compleja para datos relativamente grandes.

Los algoritmos prácticos de clustering examinan solo una parte de todas las posibles asignaciones $k = C(i)$, por lo tanto, convergen a óptimos locales.

Entre los más conocidos, tenemos

- $k-$means
- $k-$medoids
- fuzzy $k-$means

Clustering basado en algoritmos combinatorios

La solución del problema de optimización anterior usando algoritmos combinatorios puede ser computacionalmente muy compleja para datos relativamente grandes.

Los algoritmos prácticos de clustering examinan solo una parte de todas las posibles asignaciones $k = C(i)$, por lo tanto, convergen a óptimos locales.

Entre los más conocidos, tenemos

- $k-$means
- $k-$medoids
- fuzzy $k-$means

Clustering basado en algoritmos combinatorios

$k-$**means**.

Para datos cuantitativos (variables contínuas), usando la distancia euclideana como medida de disimilaridad:

$$d_{ij} = \|\mathbf{x}_i - \mathbf{x}_j\|^2.$$

En este caso,

$$W(C) = \frac{1}{2} \sum_{k=1}^{K} \sum_{C(i)=k} \sum_{C(j)=k} \|\mathbf{x}_i - \mathbf{x}_j\|^2.$$

Clustering basado en algoritmos combinatorios

$k-$**means**.

Define $N_k = \sum_{i=1}^{n} I(C(i) = k)$.
Multiplicando $W(C)$ por N_k/N_k y haciendo un poco de álgebra, obtenemos:

$$W(C) = \sum_{k=1}^{K} N_k \sum_{C(i)=k} \|\mathbf{x}_i - \bar{\mathbf{x}}_k\|^2.$$

con $\bar{\mathbf{x}}_k$ el centroide del cluster k.

Entonces, minimizar $W(C)$ es equivalente a *minimizar la varianza de cada clúster*

Clustering basado en algoritmos combinatorios

$k-$**means**.

Define $N_k = \sum_{i=1}^{n} I(C(i) = k)$.
Multiplicando $W(C)$ por N_k/N_k y haciendo un poco de álgebra, obtenemos:

$$W(C) = \sum_{k=1}^{K} N_k \sum_{C(i)=k} \|\mathbf{x}_i - \bar{\mathbf{x}}_k\|^2.$$

con $\bar{\mathbf{x}}_k$ el centroide del cluster k.

Entonces, minimizar $W(C)$ es equivalente a *minimizar la varianza de cada clúster*

Clustering basado en algoritmos combinatorios

$k-$**means**.

Una característica computacional.
Observa que, dado un conjunto S de observaciones:

$$\bar{\mathbf{x}}_s = \min_{\mathbf{m}} \sum_{i \in S} \|\mathbf{x}_i - \mathbf{m}\|^2,$$

entonces,

$$C^* = \min_{C, \{\mathbf{m}\}_1^K} \sum_{k=1}^{K} N_k \sum_{C(i)=k} \|\mathbf{x}_i - \mathbf{m}_k\|^2.$$

Esto permite hacer la minimización en dos pasos.

Clustering basado en algoritmos combinatorios

$k-$**means**.

Algoritmo:

1: Input: datos $\{\mathbf{x}_i\}_{i=1}^n$, K número de clústers
2: Proponer $k = 1, \ldots, K$ centroides de clústers $\mathbf{m}_k$
3: Minimizar la varianza intra-cluster asignando cada observación a su cluster según la distancia a su centroide:

$$C(i) = \arg \min_{1 \leq k \leq K} ||\mathbf{x}_i - \mathbf{m}_k||^2$$

4: Repetir 2-3 hasta que no haya cambios en las asignaciones

Un inconveniente: es poco robusto a datos atípicos.

Clustering basado en algoritmos combinatorios

$k-$**means**.

Algoritmo:

1: Input: datos $\{\mathbf{x}_i\}_{i=1}^n$, K número de clústers
2: Proponer $k = 1, \ldots, K$ centroides de clústers $\mathbf{m}_k$
3: Minimizar la varianza intra-cluster asignando cada observación a su cluster según la distancia a su centroide:

$$C(i) = \arg \min_{1 \leq k \leq K} ||\mathbf{x}_i - \mathbf{m}_k||^2$$

4: Repetir 2-3 hasta que no haya cambios en las asignaciones

Un inconveniente: es poco robusto a datos atípicos.

Clustering basado en algoritmos combinatorios

Gráfico de siluetas (Silhouette plot, Rousseeuw, 1987).

Nos proporciona una forma de verificar o "validar" el agrupamiento realizado basado en las proximidades o distancias dentro y entre clusters.
Supon que tenemos cierto agrupamiento C_K. Considera el encoder $C(i)$ que usamos antes, y que asigna algún cluster k al objeto $\mathbf{x}_i$. Definimos ahora

- a_i como la disimilaridad promedio del objeto $\mathbf{x}_i$ a los restantes objetos dentro de **el mismo** clúster $C(i)$
- d_i^C como la disimilaridad promedio del objeto $\mathbf{x}_i$ a los objetos que están en clústers **diferentes** de su mismo clúster, es decir, que pertenecen a algún clúster $C \neq C(i)$. Tendremos entonces $k-1$ valores diferentes para d_i^C
- $b_i = \min\limits_{c \neq C(i)} d_i^C$, en este caso, $b_i = d_i^C$ indica que el clúster C es el "vecino" del objeto $\mathbf{x}_i$, o su segunda mejor opción.

Clustering basado en algoritmos combinatorios

Gráfico de siluetas (Silhouette plot, Rousseeuw, 1987).

El valor silueta para $\mathbf{x}_i$ dado el agrupamiento C_K es

$$s_i(C_K) = s_{iK} = \frac{b_i - a_i}{\max(a_i, b_i)} \in [-1, 1]$$

- Valores cercanos a 1 indican que $\mathbf{x}_i$ está bien asignado (o clusterizado), ya que $a_i \approx 0$
- Valores cercanos a -1 indican que $\mathbf{x}_i$ tiene una clusterización deficiente, ya que $b_i \approx 0$
- $s_{iK} \approx 0$ indica que $\mathbf{x}_i$ está entre dos clusters.

Clustering basado en algoritmos combinatorios

Gráfico de siluetas (Silhouette plot, Rousseeuw, 1987).

- Definimos también, la silueta promedio $\bar{s}_K$, que es el promedio de todas las s_{iK}.
- Una forma de escoger el número de clústers, es encontrar K tal que maximice $\bar{s}_K$, para un conjunto $K = 1, 2, \ldots$ determinado.
- El coeficiente silueta $SC = \arg\max_K(\bar{s}_K)$ puede ayudarnos a hacer un diagnóstico del agrupamiento. Según Rousseeuw:

SC	Interpretación
0.71 a 1	Estructura fuerte
0.51 a 0.7	Estructura razonable
0.26 a 0.5	Estructura débil
≤ 0.25	No hay estructura

Clustering basado en algoritmos combinatorios

Código

`notebooks/clustering2.ipynb`

Clustering basado en algoritmos combinatorios

$k-$medoids

- En lugar de tomar centroides como el dato representativo de cada clúster, este método usa *algún otro dato representativo*
- Esto permite
  - usar otra medida de similaridad, ya que se optimiza explícitamente sobre las $\mathbf{m}$'s
  - usar criterios mas informativos para elegir los clusters
- En su forma mas simple, el "medoide" es un objeto del cluster que minimiza cierta función de disimilaridad

Clustering basado en algoritmos combinatorios

$k-$medoids Algoritmo:

1: Input: datos $\{\mathbf{x}_i\}_{i=1}^n$, K número de clústers
2: Asignar K clústers iniciales
3: Para alguna asignación C de clusters, encontrar la observación que resuelva

$$i_k^* = \min_{i:C(i)=k} \sum_{C(j)=k} D(\mathbf{x}_i, \mathbf{x}_j)$$

4: Asignar $\mathbf{m}_k = \mathbf{x}_{i_k^*}$.
5: Dado un conjunto de centros de clusters $\mathbf{m}_1, \dots, \mathbf{m}_K$, minimizar la disimilaridad asignando cada observación a su medoide mas cercano

$$C(i) = \arg \min_{1 \leq k \leq K} D(\mathbf{x}_i, \mathbf{m}_k)$$

6: Repetir 3-5 hasta que no haya mas cambios significativos

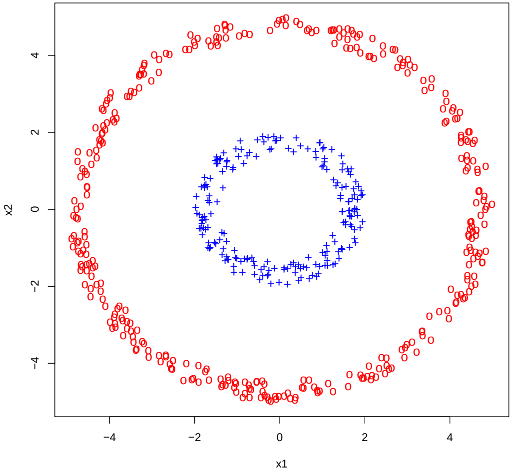
Clustering basado en algoritmos combinatorios

fuzzy $k-$means, (Kaufman and Rousseeuw (1990))

- También llamado *soft*-k means.
- Los objetos tienen una probabilidad de pertenecer a cada clúster k
- Los detalles del método, en clase...

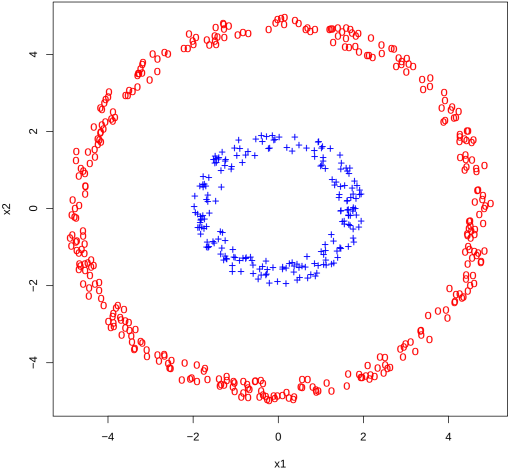
Clustering

Ahora, considera los siguientes datos para agrupar:



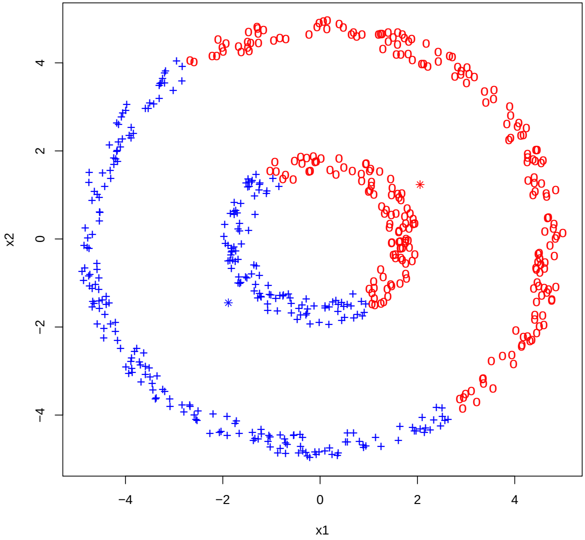
Clustering

Ahora, considera los siguientes datos para agrupar:

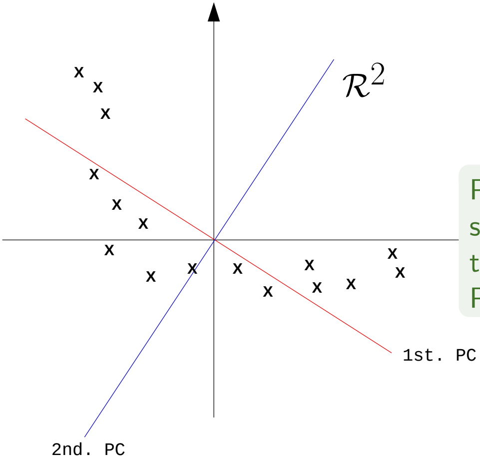


Clustering

Resultado $k-$means



Clustering



Pero éste problema, surge en otros contextos también... por ejemplo, PCA.

Clustering

what if... te digo que puedo obtener esto: