

# Aprendizaje de máquinas

## Aprendizaje no supervisado: clustering

Felipe Tobar

Facultad de Ciencias Físicas y Matemáticas  
Universidad de Chile

Otoño, 2021.

Uno de los problemas principales del aprendizaje no supervisado es poder agrupar las entradas sin necesidad de que estas tengan etiquetas (como ocurre en los métodos de clasificación).

Uno de los problemas principales del aprendizaje no supervisado es poder agrupar las entradas sin necesidad de que estas tengan etiquetas (como ocurre en los métodos de clasificación).

La dificultad principal de esta tarea es que es necesario definir un concepto de similitud entre las distintas entradas para poder particionar los datos en grupos (clusters) de características similares.

## Hierarchical clustering (HCA)

Corresponde al algoritmo de clustering más simple pero a su vez, es uno de los más caros computacionalmente. El objetivo es identificar  $k$  clusters en los datos.

# Hierarchical clustering (HCA)

Corresponde al algoritmo de clustering más simple pero a su vez, es uno de los más caros computacionalmente. El objetivo es identificar  $k$  clusters en los datos.

El algoritmo aglomerativo de clustering es el siguiente:

- 1 Para comenzar, se considerarán  $n$  clusters distintos. Asignar a cada elemento un cluster único.

# Hierarchical clustering (HCA)

Corresponde al algoritmo de clustering más simple pero a su vez, es uno de los más caros computacionalmente. El objetivo es identificar  $k$  clusters en los datos.

El algoritmo aglomerativo de clustering es el siguiente:

- ➊ Para comenzar, se considerarán  $n$  clusters distintos. Asignar a cada elemento un cluster único.
- ➋ Buscar el par de clusters más similar (bajo algún criterio) y combinarlos en un solo cluster.

# Hierarchical clustering (HCA)

Corresponde al algoritmo de clustering más simple pero a su vez, es uno de los más caros computacionalmente. El objetivo es identificar  $k$  clusters en los datos.

El algoritmo aglomerativo de clustering es el siguiente:

- ➊ Para comenzar, se considerarán  $n$  clusters distintos. Asignar a cada elemento un cluster único.
- ➋ Buscar el par de clusters más similar (bajo algún criterio) y combinarlos en un solo cluster.
- ➌ Repetir el paso anterior hasta tener  $k$  clusters.

# Hierarchical clustering (HCA)

Corresponde al algoritmo de clustering más simple pero a su vez, es uno de los más caros computacionalmente. El objetivo es identificar  $k$  clusters en los datos.

El algoritmo aglomerativo de clustering es el siguiente:

- ➊ Para comenzar, se considerarán  $n$  clusters distintos. Asignar a cada elemento un cluster único.
- ➋ Buscar el par de clusters más similar (bajo algún criterio) y combinarlos en un solo cluster.
- ➌ Repetir el paso anterior hasta tener  $k$  clusters.

Para dos clusters  $A, B \subset \mathcal{D}$ , los criterios de similitud más frecuentes son los siguientes:

- **Single-linkage clustering:**  $D_s(A, B) := \min\{d(a, b) : a \in A, b \in B\}.$



# Hierarchical clustering (HCA)

Corresponde al algoritmo de clustering más simple pero a su vez, es uno de los más caros computacionalmente. El objetivo es identificar  $k$  clusters en los datos.

El algoritmo aglomerativo de clustering es el siguiente:

- ➊ Para comenzar, se considerarán  $n$  clusters distintos. Asignar a cada elemento un cluster único.
- ➋ Buscar el par de clusters más similar (bajo algún criterio) y combinarlos en un solo cluster.
- ➌ Repetir el paso anterior hasta tener  $k$  clusters.

Para dos clusters  $A, B \subset \mathcal{D}$ , los criterios de similitud más frecuentes son los siguientes:

- **Single-linkage clustering:**  $D_s(A, B) := \min\{d(a, b) : a \in A, b \in B\}$ .
- **Complete-linkage clustering:**  $D_s(A, B) := \max\{d(a, b) : a \in A, b \in B\}$ .

# Hierarchical clustering (HCA)

Corresponde al algoritmo de clustering más simple pero a su vez, es uno de los más caros computacionalmente. El objetivo es identificar  $k$  clusters en los datos.

El algoritmo aglomerativo de clustering es el siguiente:

- 1 Para comenzar, se considerarán  $n$  clusters distintos. Asignar a cada elemento un cluster único.
- 2 Buscar el par de clusters más similar (bajo algún criterio) y combinarlos en un solo cluster.
- 3 Repetir el paso anterior hasta tener  $k$  clusters.

Para dos clusters  $A, B \subset \mathcal{D}$ , los criterios de similitud más frecuentes son los siguientes:

- **Single-linkage clustering:**  $D_s(A, B) := \min\{d(a, b) : a \in A, b \in B\}.$
- **Complete-linkage clustering:**  $D_s(A, B) := \max\{d(a, b) : a \in A, b \in B\}.$
- **Average-linkage clustering:**  $D_a(A, B) := \frac{1}{|A| \cdot |B|} \sum_{a \in A, b \in B} d(a, b).$

# Hierarchical clustering (HCA)

Corresponde al algoritmo de clustering más simple pero a su vez, es uno de los más caros computacionalmente. El objetivo es identificar  $k$  clusters en los datos.

El algoritmo aglomerativo de clustering es el siguiente:

- ➊ Para comenzar, se considerarán  $n$  clusters distintos. Asignar a cada elemento un cluster único.
- ➋ Buscar el par de clusters más similar (bajo algún criterio) y combinarlos en un solo cluster.
- ➌ Repetir el paso anterior hasta tener  $k$  clusters.

Para dos clusters  $A, B \subset \mathcal{D}$ , los criterios de similitud más frecuentes son los siguientes:

- **Single-linkage clustering:**  $D_s(A, B) := \min\{d(a, b) : a \in A, b \in B\}$ .
- **Complete-linkage clustering:**  $D_s(A, B) := \max\{d(a, b) : a \in A, b \in B\}$ .
- **Average-linkage clustering:**  $D_a(A, B) := \frac{1}{|A| \cdot |B|} \sum_{a \in A, b \in B} d(a, b)$ .

Donde  $d : \mathcal{D} \times \mathcal{D} \rightarrow \mathbb{R}_+$  es una métrica en  $\mathcal{D}$ . Elecciones distintas del criterio de similitud y/o métrica (generalmente euclidiana) pueden llevar a agrupaciones distintas.

Otro método de clustering muy utilizado es  $k$ -means. Dado  $k \in \mathbb{N}$  y un conjunto de observaciones  $X = \{x_i\}_{i=1}$  con  $x_i \in \mathbb{R}^D$  se busca separar los datos en  $k$  grupos tal que:

- A cada grupo se le asigna un centroide  $\mu_k$ .

Otro método de clustering muy utilizado es  $k$ -means. Dado  $k \in \mathbb{N}$  y un conjunto de observaciones  $X = \{x_i\}_{i=1}$  con  $x_i \in \mathbb{R}^D$  se busca separar los datos en  $k$  grupos tal que:

- A cada grupo se le asigna un centroide  $\mu_k$ .
- A cada elemento  $x_i$  se le asigna el grupo que tenga el centroide más cercano.

Otro método de clustering muy utilizado es *k*-means. Dado  $k \in \mathbb{N}$  y un conjunto de observaciones  $X = \{x_i\}_{i=1}$  con  $x_i \in \mathbb{R}^D$  se busca separar los datos en  $k$  grupos tal que:

- A cada grupo se le asigna un centroide  $\mu_k$ .
- A cada elemento  $x_i$  se le asigna el grupo que tenga el centroide más cercano.

Sea  $r_{ik}$  la asignación, esta estará definida por:

$$r_{ik} = \begin{cases} 1 & \text{si } k = \operatorname{argmin} ||x_i - x_k|| \\ 0 & \text{si no.} \end{cases}$$

Otro método de clustering muy utilizado es *k*-means. Dado  $k \in \mathbb{N}$  y un conjunto de observaciones  $X = \{x_i\}_{i=1}$  con  $x_i \in \mathbb{R}^D$  se busca separar los datos en  $k$  grupos tal que:

- A cada grupo se le asigna un centroide  $\mu_k$ .
- A cada elemento  $x_i$  se le asigna el grupo que tenga el centroide más cercano.

Sea  $r_{ik}$  la asignación, esta estará definida por:

$$r_{ik} = \begin{cases} 1 & \text{si } k = \operatorname{argmin} ||x_i - x_k|| \\ 0 & \text{si no.} \end{cases}$$

Es decir, para encontrar los centroides se debe minimizar el funcional:

$$J = \sum_{i=1}^N \sum_{k=1}^K r_{ik} ||x_i - x_k||^2$$

El algoritmo más frecuente usado para  $k$ -means es el algoritmo de Lloyd:



El algoritmo más frecuente usado para k-means es el algoritmo de Lloyd:

- **E-step:** En este paso, se calculan (actualizan) las asignaciones  $r_{ik}$ , dejando fijos  $\mu_k$ . Lo que corresponde a asignar el dato  $x_i$  al centroide más cercano.

El algoritmo más frecuente usado para k-means es el algoritmo de Lloyd:

- **E-step:** En este paso, se calculan (actualizan) las asignaciones  $r_{ik}$ , dejando fijos  $\mu_k$ . Lo que corresponde a asignar el dato  $x_i$  al centroide más cercano.
- **M-step:** El siguiente paso corresponde a actualizar los centroides  $\mu_k$  dejando fijo las asignaciones  $r_{ik}$ .

El algoritmo más frecuente usado para k-means es el algoritmo de Lloyd:

- **E-step:** En este paso, se calculan (actualizan) las asignaciones  $r_{ik}$ , dejando fijos  $\mu_k$ . Lo que corresponde a asignar el dato  $x_i$  al centroide más cercano.
- **M-step:** El siguiente paso corresponde a actualizar los centroides  $\mu_k$  dejando fijo las asignaciones  $r_{ik}$ .

Como J es cuadrática en  $\mu_k$ , entonces podemos utilizar la condición de primer orden:

$$\mu_k = \frac{\sum_{i=1}^N r_{ik} x_i}{\sum_{i=1}^N r_{ik}}$$

Lo que corresponde a asignar el centro del cluster al promedio de todas las muestras asignadas al antiguo cluster.

El algoritmo más frecuente usado para k-means es el algoritmo de Lloyd:

- **E-step:** En este paso, se calculan (actualizan) las asignaciones  $r_{ik}$ , dejando fijos  $\mu_k$ . Lo que corresponde a asignar el dato  $x_i$  al centroide más cercano.
- **M-step:** El siguiente paso corresponde a actualizar los centroides  $\mu_k$  dejando fijo las asignaciones  $r_{ik}$ .

Como  $J$  es cuadrática en  $\mu_k$ , entonces podemos utilizar la condición de primer orden:

$$\mu_k = \frac{\sum_{i=1}^N r_{ik} x_i}{\sum_{i=1}^N r_{ik}}$$

Lo que corresponde a asignar el centro del cluster al promedio de todas las muestras asignadas al antiguo cluster.

- El algoritmo termina cuando los centroides ya no cambian.

## $k$ -means: algoritmo de Lloyd

Para los centroides iniciales, se tienen dos posibles inicializaciones:

- **Método de Forgy:** se eligen de forma aleatoria  $k$  puntos de la muestra como centroides iniciales.

## $k$ -means: algoritmo de Lloyd

Para los centroides iniciales, se tienen dos posibles inicializaciones:

- **Método de Forgy:** se eligen de forma aleatoria  $k$  puntos de la muestra como centroides iniciales.
- **Random partition:** se eligen asignaciones aleatorias para los elementos. De este modo, los centroides iniciales serán los centroides obtenidos al realizar M-step.

## $k$ -means: algoritmo de Lloyd

Para los centroides iniciales, se tienen dos posibles inicializaciones:

- **Método de Forgy:** se eligen de forma aleatoria  $k$  puntos de la muestra como centroides iniciales.
- **Random partition:** se eligen asignaciones aleatorias para los elementos. De este modo, los centroides iniciales serán los centroides obtenidos al realizar M-step.

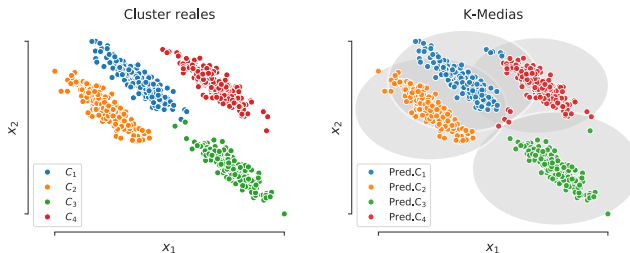
El método de Forgy es preferido cuando se realiza  $k$ -means mediante el algoritmo de Lloyd.

## k-means: algoritmo de Lloyd

Para los centroides iniciales, se tienen dos posibles inicializaciones:

- **Método de Forgy:** se eligen de forma aleatoria  $k$  puntos de la muestra como centroides iniciales.
- **Random partition:** se eligen asignaciones aleatorias para los elementos. De este modo, los centroides iniciales serán los centroides obtenidos al realizar M-step.

El método de Forgy es preferido cuando se realiza k-means mediante el algoritmo de Lloyd.



**Fig. 1.** Se observa que los clusters creados por kmeans son circulares, esto es debido a que se utiliza distancia euclidiana hacia el centro del cluster.



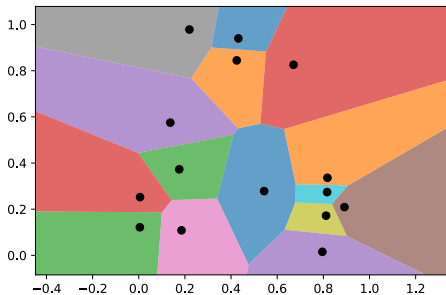
## $k$ -means: diagrama de Voronoi

La asignación de cluster mediante el centroide más cercano provoca que cada par de centroides divida al espacio ambiente en dos semiespacios mediante el hiperplano simetral que pasa entre ambos puntos.

## k-means: diagrama de Voronoi

La asignación de cluster mediante el centroide más cecano provoca que cada par de centroides divida al espacio ambiente en dos semiespacios mediante el hiperplano simetral que pasa entre ambos puntos.

Dado que la intersección finita de semiespacios genera un poliedro, se tiene que la partición generada por kmeans forma un diagrama de Voronoi:



**Fig. 2.** Partición de  $\mathbb{R}^2$  inducida por los centroides.

## Modelo de mezcla de gaussianas (GMM)

La mezcla de gaussianas (GMM) es un caso general de  $k$ -means, en donde los clusters pueden tener una forma anisotrópica modelada por una Gaussiana. Si bien su solución puede ser muy similar a  $k$ -means los supuestos subyacentes al modelo GMM son diferentes y obedecen a un enfoque de modelo generativo.

## Modelo de mezcla de gaussianas (GMM)

La mezcla de gaussianas (GMM) es un caso general de  $k$ -means, en donde los clusters pueden tener una forma anisotrópica modelada por una Gaussiana. Si bien su solución puede ser muy similar a  $k$ -means los supuestos subyacentes al modelo GMM son diferentes y obedecen a un enfoque de modelo generativo.

Una distribución de mezcla de gaussianas consiste en una combinación convexa de distribuciones gaussianas

$$p(\mathbf{x}) = \sum_{k=1}^K \pi_k \mathcal{N}(\mathbf{x} | \mu_k, \Sigma_k)$$

Donde una muestra  $x$  es generada mediante dos etapas: primero se elige un cluster al azar (de acuerdo a  $\pi_k$ ) y luego, se genera una muestra aleatoria dentro del cluster.

## Modelo de mezcla de gaussianas (GMM)

La mezcla de gaussianas (GMM) es un caso general de  $k$ -means, en donde los clusters pueden tener una forma anisotrópica modelada por una Gaussiana. Si bien su solución puede ser muy similar a  $k$ -means los supuestos subyacentes al modelo GMM son diferentes y obedecen a un enfoque de modelo generativo.

Una distribución de mezcla de gaussianas consiste en una combinación convexa de distribuciones gaussianas

$$p(\mathbf{x}) = \sum_{k=1}^K \pi_k \mathcal{N}(\mathbf{x} | \mu_k, \Sigma_k)$$

Donde una muestra  $x$  es generada mediante dos etapas: primero se elige un cluster al azar (de acuerdo a  $\pi_k$ ) y luego, se genera una muestra aleatoria dentro del cluster. Nos referiremos a los parámetros de este modelo como

- $\pi_k$  : coeficiente de mezcla del cluster  $k$  (probabilidad de venir del cluster  $k$ ).
- $\mu_k$  : media del cluster  $k$ .
- $\Sigma_k$  : matriz de covarianza del cluster  $k$ .

## Modelo de mezcla de gaussianas (GMM): MV

La log-verosimilitud del modelo está dada por

$$\log p(\mathbf{x}_1, \dots, \mathbf{x}_N) = \sum_{n=1}^N \log p(\mathbf{x}_n) = \sum_{n=1}^N \log \sum_{k=1}^K \pi_k \mathcal{N}(\mathbf{x}_n | \mu_k, \Sigma_k)$$

## Modelo de mezcla de gaussianas (GMM): MV

La log-verosimilitud del modelo está dada por

$$\log p(\mathbf{x}_1, \dots, \mathbf{x}_N) = \sum_{n=1}^N \log p(\mathbf{x}_n) = \sum_{n=1}^N \log \sum_{k=1}^K \pi_k \mathcal{N}(\mathbf{x}_n | \mu_k, \Sigma_k)$$

Hay una serie de complicaciones relacionados a la búsqueda de estos parámetros mediante MV:

- **Singularidades:** cuando una muestra es exactamente igual a una de las medias, en cuyo caso un término de la log-verosimilitud es proporcional a  $1/\sigma_k$ , con lo que la maximización de  $\sigma_k$  resulta en una log-verosimilitud infinita.

## Modelo de mezcla de gaussianas (GMM): MV

La log-verosimilitud del modelo está dada por

$$\log p(\mathbf{x}_1, \dots, \mathbf{x}_N) = \sum_{n=1}^N \log p(\mathbf{x}_n) = \sum_{n=1}^N \log \sum_{k=1}^K \pi_k \mathcal{N}(\mathbf{x}_n | \mu_k, \Sigma_k)$$

Hay una serie de complicaciones relacionados a la búsqueda de estos parámetros mediante MV:

- **Singularidades:** cuando una muestra es exactamente igual a una de las medias, en cuyo caso un término de la log-verosimilitud es proporcional a  $1/\sigma_k$ , con lo que la maximización de  $\sigma_k$  resulta en una log-verosimilitud infinita.
- **Redundancias de soluciones:** para cada máximo local de la log-verosimilitud existen  $k!$  soluciones equivalente con la misma verosimilitud dadas por las permutaciones de las etiquetas de los clusters.



# Modelo de mezcla de gaussianas (GMM): MV

La log-verosimilitud del modelo está dada por

$$\log p(\mathbf{x}_1, \dots, \mathbf{x}_N) = \sum_{n=1}^N \log p(\mathbf{x}_n) = \sum_{n=1}^N \log \sum_{k=1}^K \pi_k \mathcal{N}(\mathbf{x}_n | \mu_k, \Sigma_k)$$

Hay una serie de complicaciones relacionados a la búsqueda de estos parámetros mediante MV:

- **Singularidades:** cuando una muestra es exactamente igual a una de las medias, en cuyo caso un término de la log-verosimilitud es proporcional a  $1/\sigma_k$ , con lo que la maximización de  $\sigma_k$  resulta en una log-verosimilitud infinita.
- **Redundancias de soluciones:** para cada máximo local de la log-verosimilitud existen  $k!$  soluciones equivalente con la misma verosimilitud dadas por las permutaciones de las etiquetas de los clusters.
- **Funcional complejo:** la sumatoria aparece *dentro* del logaritmo, consecuentemente, el logaritmo no actúa directamente en la gaussiana reduciendo el funcional de optimización a una solución en forma cerrada.

Este último problema obliga a considerar métodos basados en gradientes.

## Modelo de mezcla de gaussianas (GMM): entrenamiento

Veamos las condiciones de primer orden sobre la log-verosimilitud para encontrar los parámetros.

Denotando  $\gamma(z_k(\mathbf{x}_i)) = p(z_k(\mathbf{x}_i) = 1 | \mathbf{x}_i)$ , se tiene el siguiente resultado:

### Lemma

*Para el modelo GGM, los parámetros óptimos son*

$$\mu_k = \frac{1}{R_k} \sum_{i=1}^N \gamma(z_k(\mathbf{x}_i)) \mathbf{x}_i$$

$$\Sigma_k = \frac{1}{R_k} \sum_{i=1}^N \gamma(z_k(\mathbf{x}_i)) (\mathbf{x}_i - \mu_k)(\mathbf{x}_i - \mu_k)^\top$$

$$\pi_k = \frac{R_k}{R}$$

donde  $R_k = \sum_{i=1}^N \gamma(z_k(\mathbf{x}_i))$  y  $R = \sum_{k=1}^K R_k$ .

Esto no constituye una solución en forma cerrada para los parámetros ya que la posterior  $\gamma(z_k(\mathbf{x}_i))$  depende de todos los parámetros:

$$p(z_k(\mathbf{x}) = 1|\mathbf{x}) = \frac{p(\mathbf{x}|z_k(\mathbf{x}) = 1)p(z_k(\mathbf{x}) = 1)}{p(\mathbf{x})} = \frac{\pi_k \mathcal{N}(\mathbf{x}|\mu_k, \Sigma_k)}{\sum_{k=1}^K \pi_k \mathcal{N}(\mathbf{x}|\mu_k, \Sigma_k)}$$

Esto no constituye una solución en forma cerrada para los parámetros ya que la posterior  $\gamma(z_k(\mathbf{x}_i))$  depende de todos los parámetros:

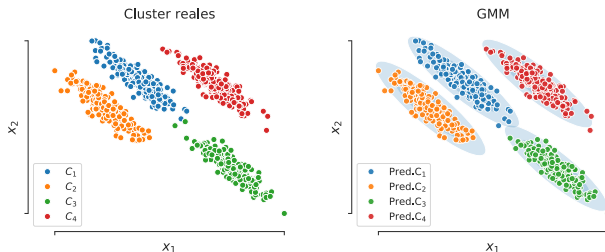
$$p(z_k(\mathbf{x}) = 1|\mathbf{x}) = \frac{p(\mathbf{x}|z_k(\mathbf{x}) = 1)p(z_k(\mathbf{x}) = 1)}{p(\mathbf{x})} = \frac{\pi_k \mathcal{N}(\mathbf{x}|\mu_k, \Sigma_k)}{\sum_{k=1}^K \pi_k \mathcal{N}(\mathbf{x}|\mu_k, \Sigma_k)}$$

Sin embargo, podemos considerar un procedimiento iterativo en donde calculamos las posteriores  $\gamma(z_k(\mathbf{x}_i))$  (llamado paso E), para luego calcular los parámetros óptimos de acuerdo a las ecuaciones anteriores (llamado paso M).

Esto no constituye una solución en forma cerrada para los parámetros ya que la posterior  $\gamma(z_k(\mathbf{x}_i))$  depende de todos los parámetros:

$$p(z_k(\mathbf{x}) = 1|\mathbf{x}) = \frac{p(\mathbf{x}|z_k(\mathbf{x}) = 1)p(z_k(\mathbf{x}) = 1)}{p(\mathbf{x})} = \frac{\pi_k \mathcal{N}(\mathbf{x}|\mu_k, \Sigma_k)}{\sum_{k=1}^K \pi_k \mathcal{N}(\mathbf{x}|\mu_k, \Sigma_k)}$$

Sin embargo, podemos considerar un procedimiento iterativo en donde calculamos las posteriores  $\gamma(z_k(\mathbf{x}_i))$  (llamado paso E), para luego calcular los parámetros óptimos de acuerdo a las ecuaciones anteriores (llamado paso M).



**Fig. 3.** Se observa como GMM es una generalización de  $k$ -means, donde ahora los clusters tienen forma de gaussiana anisotrópica.