

# Agrupamiento y clasificación en la recuperación de información en la web



Integrantes:

Marcos Manuel Tirador del Riego

Laura Victoria Riera Pérez

Tercer año. Ciencias de la computación. Universidad de La Habana. Cuba.

Noviembre, 2022

# Índice general I

## 1 Agrupamiento

- Medidas de similitud entre documentos

- Medidas de evaluación

- Agrupamiento particionado

- Agrupamiento jerárquico

  - Agrupamiento jerárquico aglomerativo

  - Agrupamiento jerárquico divisivo

- Ventajas

- Desventajas

- Ejemplos de aplicación

## Índice general II

## 2 Clasificación

## Naive Bayes

## Feature Selection

## K Nearest Neighbor

## Medidas de evaluación

## Ventajas

## Desventajas

## Aplicaciones en la Recuperación de la Información

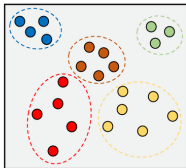
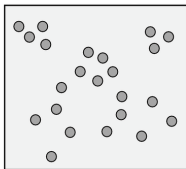
## Otros ejemplos de aplicación

### 3 Conclusiones

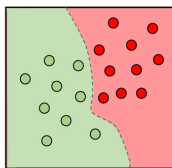
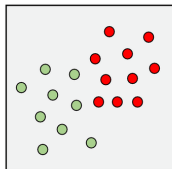
#### 4 Referencias

# Aprendizaje no supervisado vs. aprendizaje supervisado

Aprendizaje no  
supervisado

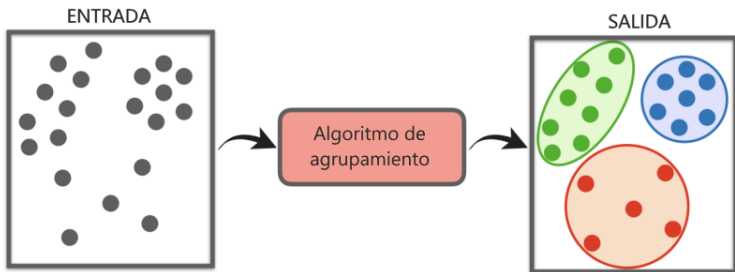


Aprendizaje  
supervisado



# Agrupamiento I

Los algoritmos de agrupamiento conglomeran un conjunto de documentos en subconjuntos o clústeres. Son utilizados para generar una estructura de categorías que se ajuste a un conjunto de observaciones.





## Hipótesis de agrupamiento

*"Los documentos en el mismo grupo se comportan de manera similar con respecto a la relevancia para las necesidades de información."*

La hipótesis establece que si hay un documento de un grupo que es relevante a una solicitud de búsqueda, entonces es probable que otros documentos del mismo clúster también sean relevantes.

# Clasificación de los algoritmos de agrupamiento

Según la pertenencia a los grupos:

- *agrupamiento exclusivo o fuerte* (hard clustering): cada documento es miembro de exactamente un grupo.
- *agrupamiento difuso o suave* (soft clustering): un documento tiene membresía fraccionaria en varios grupos.

Según el tipo de estructura impuesta sobre los datos:

- *agrupamiento particionado o plano* (flat clustering)
- *agrupamiento difuso o suave* (soft clustering): *agrupamiento jerárquico* (hierarchical clustering).



## Medidas de similitud

Sean  $d_i$  el documento  $i$  del corpus y  $w_{ik}$  el peso del término  $k$  de un total  $N$  ( $N > 0$ ) en el documento  $i$ .

- **Coeficiente de Dice:**

$$S_{d_i, d_j} = \frac{2 \sum_{k=1}^N (w_{ik} w_{jk})}{\sum_{k=1}^N w_{ik}^2 + \sum_{k=1}^N w_{jk}^2}$$

- **Coeficiente de Jaccard:**

$$S_{d_i, d_j} = \frac{\sum_{k=1}^N (w_{ik} w_{jk})}{\sum_{k=1}^N w_{ik}^2 + \sum_{k=1}^{max} w_{jk}^2 - \sum_{k=1}^N (w_{ik} w_{jk})}$$

- **Coeficiente del coseno:**

$$S_{d_i, d_j} = \frac{\sum_{k=1}^N (w_{ik} w_{jk})}{\sqrt{\sum_{k=1}^N w_{ik}^2 \sum_{k=1}^N w_{jk}^2}}$$

# Medidas de evaluación I

- **Pureza:** Para calcular la pureza, cada grupo se asigna a la clase que es más frecuente en el grupo, y luego se mide la precisión de esta asignación contando el número de documentos correctamente asignados y dividiendo por  $N$ .

$$purity(\Omega, C) = \frac{1}{N} \sum_k \max_j |\omega_k \cap c_j|$$

donde:

- $\Omega = \{\omega_1, \omega_2, \dots, \omega_K\} \rightarrow$  conjunto de clústers
- $C = \{c_1, c_2, \dots, c_J\} \rightarrow$  conjunto de clases

Se interpreta  $\omega_k$  como el conjunto de documentos en el clúster y  $c_j$  como el conjunto de documentos que pertenecen a esa clase.

Malos agrupamientos tienen valores de pureza cercanos a 0, mientras que un agrupamiento perfecto tiene una pureza 1.

## Medidas de evaluación II

- **Índice de Rand (Rand Index):** Se deben asignar dos documentos al mismo clúster si y sólo si son similares. El índice de Rand (RI) mide esta precisión.

$$RI = \frac{TP + TN}{TP + FP + FN + TN}$$

donde:

- $TP \rightarrow$  decisión positiva verdadera (se asignan dos documentos similares al mismo grupo)
- $TN \rightarrow$  decisión negativa verdadera (se asignan dos documentos no similares a diferentes grupos)
- $FP \rightarrow$  decisión de falso positivo (se asignan dos documentos no similares al mismo grupo)
- $FN \rightarrow$  decisión de falso negativo (se asignan dos documentos similares a diferentes agrupaciones)

El índice de Rand otorga el mismo peso a los falsos positivos y falsos negativos, sin embargo, separar documentos similares a veces es peor que poner pares de dos documentos no similares en el mismo grupo.

## Medidas de evaluación III

- **Medida F:** Se puede usar la medida F, vista anteriormente en conferencia, para penalizar los falsos negativos más fuertemente que los falsos positivos seleccionando un valor  $\beta > 1$ , dando así más peso al recobrado.

$$P = \frac{TP}{TP + FP} \quad R = \frac{TP}{TP + FN}$$

$$F_{\beta} = \frac{(\beta^2 + 1)PR}{\beta^2 P + R}$$

# Agrupamiento particionado

El agrupamiento particionado crea un conjunto de clústeres sin ninguna estructura explícita que los relacione entre sí.

# K-means I

Es el algoritmo de agrupamiento plano más importante. Su *objetivo* es minimizar la distancia euclidiana al cuadrado promedio entre los documentos y el centro de sus clústeres.

El centro de un clúster se define como la media o centroide  $\mu$  de los documentos en un grupo  $\omega$ :

$$\vec{\mu}(\omega) \leftarrow \frac{1}{|\omega_k|} \sum_{\vec{x} \in \omega_k} \vec{x}$$

Se asume que los documentos se representan como vectores de longitud normalizada en un espacio de valor real de la manera habitual.

# K-means II

Una medida de qué tan bien los centroides representan a los miembros de su clúster es la suma residual de cuadrados o RSS, que es la distancia al cuadrado de cada vector desde su centroide sumado sobre todos los vectores:

$$RSS_k = \sum_{\vec{x} \in \omega_k} |\vec{x} - \vec{\mu}(\omega_k)|^2$$

$$RSS = \sum_{k=1}^K RSS_k$$

RSS es entonces la *función objetivo* en K-means y nuestro objetivo es minimizarla.

# K-means III

---

## Algorithm 1 K-Means

---

**Require:**  $\{\vec{x}_1, \dots, \vec{x}_N\}, K$

1:  $(\vec{s}_1, \dots, \vec{s}_K) \leftarrow \text{SelectRandomSeeds}(\{\vec{x}_1, \dots, \vec{x}_N\}, K)$

2: **for**  $k \leftarrow 1$  **to**  $K$  **do**

3:      $\vec{\mu}_k \leftarrow \vec{s}_k$

4: **while** stopping criterion has not been met **do**

5:     **for**  $k \leftarrow 1$  **to**  $K$  **do**

6:          $\omega_k \leftarrow \{\}$

7:     **for**  $n \leftarrow 1$  **to**  $N$  **do**

8:          $j \leftarrow \arg \min_{j'} |\vec{\mu}_{j'} - \vec{x}_n|$

9:          $\mu_j \leftarrow \omega_j \cup \{\vec{x}_n\}$  (reassignment of vectors)

10:     **for**  $k \leftarrow 1$  **to**  $K$  **do**

11:          $\vec{\mu}_k \leftarrow \frac{1}{|\omega_k|} \sum_{\vec{x} \in \omega_k} \vec{x}$  (recomputation of centroids)

12: **return**  $\{\vec{\mu}_1, \dots, \vec{\mu}_N\}$

---



# K-means IV

El primer paso de esta implementación de K-means es seleccionar al azar como centros iniciales de los clústeres a K documentos, estas son las semillas. Luego, el algoritmo mueve los centros de los grupos en el espacio para minimizar el RSS. Este proceso se repite de manera iterativa hasta que se cumpla un criterio de parada.

Criterios de parada:

- Cuando se ha completado un número fijo de iteraciones I.
- Cuando la asignación de documentos a grupos no cambia entre iteraciones, excepto en los casos con un mínimo local malo.
- Cuando los centroides no cambian entre iteraciones.
- Cuando RSS cae por debajo de un umbral.

# Agrupamiento jerárquico

El *agrupamiento jerárquico* produce una jerarquía, una estructura que es más informativa que el conjunto no estructurado de clústeres devuelto por el agrupamiento particionado, no requiere que especifiquemos previamente el número de grupos y la mayoría son deterministas, sin embargo son más ineficientes que los particionados. En una representación gráfica los elementos quedan anidados en jerarquías con forma de árbol.

Los algoritmos de agrupamiento jerárquico pueden tener dos enfoques: de arriba hacia abajo (top-down) llamados de *agrupamiento jerárquico aglomerativo* o de abajo hacia arriba (bottom-up) conocidos como de *agrupamiento jerárquico divisivo*.

# Agrupamiento jerárquico aglomerativo

Los algoritmos de abajo hacia arriba tratan cada documento como un clúster único desde el principio y luego fusionan (o aglomeran) sucesivamente pares de grupos hasta que todos los grupos se han fusionado en uno solo que contiene todos los documentos. Es por esto que se denomina agrupamiento jerárquico aglomerativo o HAC por sus siglas en inglés.

Toman decisiones basadas en patrones locales sin tener inicialmente en cuenta la distribución global. Estas decisiones tempranas no se pueden deshacer.

# Medidas de similitud para clústeres en HAC I

- **Agrupamiento por enlazamiento único** (Single link clustering): La similitud entre dos clústers es la similitud de los dos objetos más cercanos entre ellos (mayor similitud).

$$sim(\omega_i, \omega_j) = \max_{\vec{x} \in \omega_i, \vec{y} \in \omega_j} SIM(\vec{x}, \vec{y})$$

- **Agrupamiento por enlazamiento completo** (complete link clustering): La similitud entre dos clústers es la similitud de los dos objetos más alejados entre ellos (menor similitud).

$$sim(\omega_i, \omega_j) = \min_{\vec{x} \in \omega_i, \vec{y} \in \omega_j} SIM(\vec{x}, \vec{y})$$

## Medidas de similitud para clústeres en HAC II

- Agrupamiento aglomerativo por promedio de grupo**  
 (group-average agglomerative clustering): El agrupamiento aglomerativo por promedio de grupo o GAAC por sus siglas en inglés, calcula la similitud promedio SIM-GA de todos los pares de documentos, incluidos los pares del mismo grupo (las auto-similitudes no están incluidas en el promedio).

$$SIM - GA(\omega_i, \omega_j) = \frac{1}{(N_i + N_j)(N_i + N_j - 1)} \sum_{\vec{x} \in \omega_i \cup \omega_j} \sum_{\vec{y} \in \omega_i \cup \omega_j, \vec{x} \neq \vec{y}} \vec{x} \cdot \vec{y}$$

Este evalúa la calidad de un clúster basada en todas las similitudes entre documentos, evitando así castigar valores extremos como en los criterios de enlace único y enlace completo, que establecen la similitud del clúster con la similitud de un solo par de documentos.

## Medidas de similitud para clústeres en HAC III

- **Agrupamiento por centroide** (centroid clustering): La similitud de dos clústeres está definida como la similitud de sus centroides

$$\begin{aligned}
 \text{sim}(\omega_i, \omega_j) &= \mu(\omega_i) \cdot \mu(\omega_j) \\
 &= \left( \frac{1}{N_i} \sum_{\vec{x} \in \omega_i} \vec{x} \right) \cdot \left( \frac{1}{N_j} \sum_{\vec{y} \in \omega_j} \vec{y} \right) \\
 &= \frac{1}{N_i N_j} \sum_{\vec{x} \in \omega_i} \sum_{\vec{y} \in \omega_j} \vec{x} \cdot \vec{y}
 \end{aligned}$$

A diferencia de GAAC, el agrupamiento por centroide excluye el cálculo de pares del mismo clúster.

# Algoritmo HAC I

Dado un conjunto de  $N$  elementos a agrupar, el proceso básico del agrupamiento jerárquico es:

- 1 Se comienza con  $N$  clústeres, resultado de asignar cada elemento al suyo propio. Se computa la matriz  $C$  de similitud de  $N \times N$ .
- 2 Se halla la similitud entre los pares de clústeres con la medida deseada.
- 3 Se toma el par más similar de clústeres y se combinan en un único clúster.
- 4 Se calculan las similitudes entre el nuevo clúster y cada uno de los clústeres antiguos.
- 5 Se repiten los pasos 3 y 4 hasta que todos los elementos estén agrupados en un solo grupo de tamaño  $N$ .

# Algoritmo HAC II

---

## Algorithm 2 HAC

---

**Require:**  $\{d_1, \dots, d_N\}$

```

1: for  $n \leftarrow 1$  to  $N$  do
2:   for  $i \leftarrow 1$  to  $N$  do
3:      $C[n][i] \leftarrow SIM(d_n, d_i)$ 
4:      $I[n] \leftarrow 1$  (keeps track of active clusters)
5:    $A \leftarrow []$  (assembles clustering as a sequence of merges)
6: for  $k \leftarrow 1$  to  $N - 1$  do
7:    $\langle i, m \rangle \leftarrow \arg \max_{\langle i, m \rangle: i \neq m \wedge I[i]=1 \wedge I[m]=1} C[i][m]$ 
8:    $A.APPEND(\langle i, m \rangle)$  (store merge)
9:   for  $j \leftarrow 1$  to  $N$  do
10:     $C[i][j] \leftarrow SIM(i, m, j)$ 
11:     $C[j][i] \leftarrow SIM(i, m, j)$ 
12:     $I[m] \leftarrow 0$  (deactivate cluster)
13: return  $A$ 

```

---



# Algoritmo HAC III

## Notas:

- En cada iteración, los dos clústeres más similares se fusionan y las filas y columnas del clúster fusionado  $i$  en  $C$  se actualizan.
- El agrupamiento se almacena como una lista de fusiones en  $A$ .
- $I$  indica qué clústeres aún están disponibles para fusionarse.
- La función  $SIM(i, m, j)$  calcula la similitud del grupo  $j$  con la fusión de los grupos  $i$  y  $M$ .

Una suposición fundamental en HAC es que la operación de fusión es monótona, es decir si  $s_1, s_2, \dots, s_{K-1}$  son las similitudes de combinación de las fusiones sucesivas de un HAC, entonces se cumple

$$s_1 \geq s_2 \geq \dots \geq s_{K-1}.$$

# Agrupamiento jerárquico divisivo

Los algoritmos de arriba hacia abajo comienzan con todos los documentos en un grupo. El clúster se divide utilizando un algoritmo de agrupamiento particionado. Este procedimiento se aplica recursivamente hasta que cada documento está en su propio clúster.

A pesar de necesitar un segundo algoritmo de agrupamiento particionado como una subrutina, tiene la ventaja de ser más eficiente si no generamos una jerarquía completa hasta las hojas de documentos individuales. Para un número fijo de niveles, y utilizando un algoritmo particionado eficiente como K-means, los algoritmos divisivos son lineales en el número de documentos y clústeres.

Además se beneficia de la información completa sobre la distribución global al tomar decisiones de partición de alto nivel.

# Ventajas

- No es necesario identificar las clases antes del procesamiento por lo que no se debe contar con expertos para este fin.
- Es útil para proporcionar estructura en grandes conjuntos de datos multivariados.
- Se ha descrito como una herramienta de descubrimiento porque tiene el potencial para revelar relaciones previamente no detectadas basadas en datos complejos.
- Debido a su amplia aplicación en disímiles campos, cuenta el apoyo de una serie de paquetes de software, a menudo disponibles en la informática académica y otros entornos, por lo que se facilita su utilización.

# Desventajas

- No se tiene una idea exacta de las clases creadas.
- No recibe retroalimentación.

# Ejemplos de aplicación I

El agrupamiento es una técnica importante para descubrir subregiones o subespacios relativamente densos de una distribución de datos multidimensional. Se ha utilizado en la recuperación de información para muchos propósitos diferentes, como la expansión de consultas, la agrupación e indexación de documentos y la visualización de resultados de búsqueda. Permiten mejorar interfaz y experiencia de usuario y proporcionar una mayor eficacia o eficiencia del sistema de búsqueda.

## Ejemplos de aplicación II

A continuación se describen con más detalle algunas de las aplicaciones más importantes:

- Agrupamiento de resultados de búsqueda (Search result clustering):

La presentación predeterminada de los resultados de búsqueda (documentos devueltos en respuesta a una consulta) en la recuperación de información es una lista sencilla. Los usuarios escanean la lista de arriba a abajo hasta que encuentran la información que buscan.

En su lugar, en la agrupación en clústeres de resultados de búsqueda los documentos similares aparecen juntos, siendo más fácil escanear algunos grupos coherentes que muchos documentos individuales. Esto es particularmente útil si un término de búsqueda tiene diferentes significados.

## Ejemplos de aplicación III

- Dispersión-recopilación (Scatter-Gather): Su objetivo es también una mejor interfaz de usuario. Este agrupa toda la colección para obtener grupos de documentos que el usuario puede seleccionar o reunir manualmente. Los grupos seleccionados se fusionan y el conjunto resultante se vuelve a agrupar. Este proceso se repite hasta que se encuentre un grupo de interés.

La navegación basada en la agrupación de clústeres es una alternativa interesante a la búsqueda de palabras clave a la información estándar paradigma de recuperación de información. Esto es especialmente cierto en escenarios donde los usuarios prefieren navegar en lugar de buscar porque no están seguros de qué búsqueda términos a utilizar.

## Ejemplos de aplicación IV

- Modelado de lenguaje (Language modeling): Explota directamente la hipótesis del agrupamiento para mejorar los resultados de búsqueda, basado en una agrupación de toda la colección. Usamos un índice invertido estándar para identificar un conjunto inicial de documentos que coincide con la consulta, pero luego agregamos otros documentos de los mismos grupos incluso si tienen poca similitud con la consulta.

Para evitar problemas de datos escasos en el modelado lenguaje enfocado a RI, el modelo de documento  $d$  se puede interpolar con un modelo de colección. Pero la colección contiene muchos documentos con términos atípicos de  $d$ . Al reemplazar el modelo de colección con un modelo derivado de grupo de  $d$ , obtenemos estimaciones más precisas de las probabilidades de ocurrencia de términos en  $d$ .



## Ejemplos de aplicación V

- Recuperación basada en clústeres (Cluster-based retrieval): La agrupación también puede acelerar la búsqueda. La búsqueda en el modelo de espacio vectorial equivale a encontrar los vecinos más cercanos a la consulta. El índice invertido admite la búsqueda rápida del vecino más cercano para la configuración estándar de RI. Sin embargo, a veces es posible que no podamos usar un índice invertido de manera eficiente. En tales casos, podríamos calcular la similitud de la consulta con cada documento, pero esto es lento. La hipótesis de agrupamiento ofrece una alternativa: encontrar los clústeres que están más cerca de la consulta y sólo considerar los documentos de estos. Como hay muchos menos clústeres que documentos, se disminuye grandemente el espacio de búsqueda, y encontrar el clúster más cercano es rápido. Además los elementos que coinciden con una consulta son similares

# Ejemplos de aplicación VI

entre sí, por lo que tienden a estar en los mismos clústeres, de esta forma la calidad no disminuye en gran medida.





# Feature Selection

## K Nearest Neighbor









## Aplicaciones en la Recuperación de la Información

## Otros ejemplos de aplicación



