

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

- Ventajas

- Desventajas

- Ejemplos de aplicación

Índice general II

Agrupamiento jerárquico aglomerativo

Agrupamiento jerárquico divisivo

2 Clasificación

Naive Bayes

Feature Selection

K Nearest Neighbor

Medidas de evaluación

Ventajas

Desventajas

Aplicaciones en la Recuperación de la Información

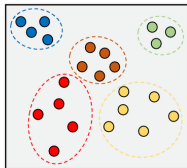
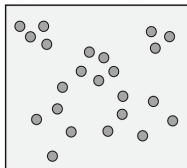
Otros ejemplos de aplicación

3 Conclusiones

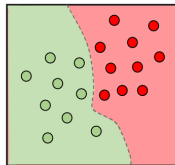
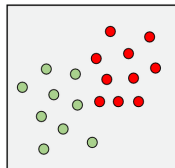
4 Referencias

Aprendizaje no supervisado vs. aprendizaje supervisado

Aprendizaje no
supervisado

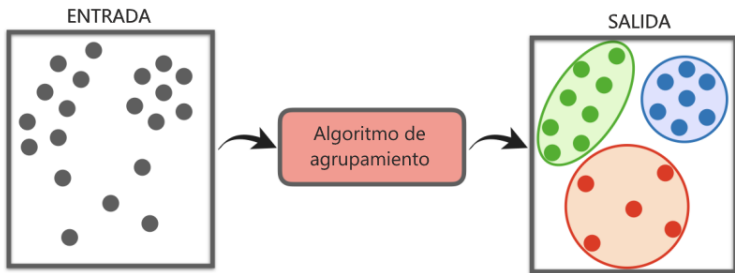


Aprendizaje
supervisado



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.



Características generales

- Es la forma más común de aprendizaje no supervisado.
- Los grupos formados deben tener un alto grado de asociación entre los documentos de un mismo grupo y un bajo grado entre miembros de diferentes grupos.
- La entrada clave para un algoritmo de agrupamiento es la medida de distancia. Diferentes medidas de distancia dan lugar a diferentes agrupamientos.

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 jerárquico* (hierarchical clustering).

Medidas de similitud

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

- **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**
- **Información mutua normalizada**
- **Índice de Rand (Rand Index)**
- **Medida F**

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 particionado 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 μ de los documentos en un grupo ω :

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

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.
- 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 clusters devuelto por el agrupamiento particionado.

Pueden tener dos enfoques:

- De abajo hacia arriba (bottom-up) llamados de *agrupamiento jerárquico aglomerativo*.
- De arriba hacia abajo (top-down) 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 clústeres es la similitud de los dos objetos más cercanos entre ellos (mayor similitud).
- **Agrupamiento por enlazamiento completo** (complete link clustering): La similitud entre dos clústeres es la similitud de los dos objetos más alejados entre ellos (menor similitud).
- **Agrupamiento aglomerativo por promedio de grupo** (group-average agglomerative clustering): Calcula la similitud promedio de todos los pares de documentos, incluidos los pares del mismo grupo, evitando así castigar valores extremos como en los criterios de enlace único y enlace completo.
- **Agrupamiento por centroide** (centroid clustering): La similitud de dos clústeres está definida como la similitud de sus centroides.

Algoritmo HAC I

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

```

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.

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 en la web 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 en la web II

- **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 clusters 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 en la web 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 clusters es una alternativa interesante a la búsqueda por palabras clave, el paradigma de recuperación de información estándar. Esto es especialmente cierto en escenarios donde los usuarios prefieren navegar en lugar de buscar porque no están seguros de qué términos utilizar en la búsqueda.

Ejemplos de aplicación en la web IV

- Recuperación basada en clústeres** (Cluster-based retrieval): 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, sin embargo, a veces es posible que no se pueda usar un índice invertido de manera eficiente. En tales casos, se podría calcular la similitud de la consulta con cada documento, pero esto es lento. La hipotesis 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. Además, los elementos que coinciden con una consulta son similares entre sí, por lo que tienden a estar en los mismos clústeres, de esta forma la calidad no disminuye en gran medida.

Clasificación I

El problema de clasificación en sentido general consiste en determinar dentro de un conjunto de clases a cuál de ellas pertenece un objeto dado. En el marco de este documento es de interés estudiar la clasificación de textos.

La forma más antigua de llevar a cabo la clasificación es manualmente. Por ejemplo, los bibliotecarios clasifican los libros de acuerdo a ciertos criterios, de modo que encontrar una información buscada no resulte una tarea de gigante dificultad. Sin embargo, la clasificación manual tiene sus límites de escalabilidad.

Clasificación II

Como alternativa podría pensarse el uso de *reglas* para determinar automáticamente si un texto pertenece o no a una clase determinada de documentos.

Se ilustrará un ejemplo de regla aplicada automáticamente. Supongamos que un usuario necesita hacer una consulta en una página de noticias, por ejemplo, necesita tener actualidad sobre noticias relacionadas con las finanzas para tomar decisiones en su negocio. Al usuario entonces le podría interesar que de hacer una consulta en el sistema dicha consulta se mantenga ejecutando y le provea periódicamente las noticias relativas a finanzas.

Clasificación III

A este tipo de consultas se le denomina consultas permanentes (o *standing queries* en inglés). Una consulta permanente es aquella que se ejecuta periódicamente en una colección a la cual nuevos documentos se adicionan en el tiempo. Toda consulta permanente se puede ver como un tipo de regla que se aplica a un sistema de clasificación que divide una colección en documentos que satisfacen la consulta y documentos que no. Una regla captura una cierta combinación de palabras claves que identifican una clase. Reglas codificadas a mano pueden llegar a ser altamente escalable, pero crearlas y mantenerlas requiere un elevado costo en recursos humanos.

Clasificación IV

Existe, sin embargo, un enfoque adicional a los dos anteriores mencionados. Este es el uso de *Aprendizaje de Máquinas*. En este enfoque el conjunto de reglas de clasificación, o en general, el criterio usado para clasificar, es aprendido de forma automática a partir de los datos de entrenamiento.

Al uso del Aprendizaje de Máquinas en la clasificación de textos se le conoce como clasificación estadística de texto (o en inglés *statistical text classification*) si el método de aprendizaje usado es estadístico.

Clasificación V

Se presenta a continuación la definición formal del problema de clasificación de textos, en el contexto del Aprendizaje de Máquinas.

Definición: Sea \mathcal{X} el espacio de documentos y $\mathcal{C} := \{c_i \mid c_i \subset \mathcal{X}, i \in \{1, 2, \dots, n\}\}$ un conjunto fijo de clases (también llamadas categorías o etiquetas). Sea además D un conjunto entrenado de documentos clasificados $(d, c) \in \mathcal{X} \times \mathcal{C}$. El *problema de la clasificación de textos* consiste en encontrar, usando métodos o algoritmos de aprendizaje, una función *clasificadora* $\gamma: \mathcal{X} \rightarrow \mathcal{C}$, que mapee documentos a clases, que satisfaga que $D \subset \gamma$.

Clasificación VI

El aprendizaje que toma parte en la búsqueda de γ es llamado *aprendizaje supervisado* debido a que se necesita la ayuda de uno o varios expertos que creen el conjunto de entrenamiento D . Estos expertos son también quienes determinan el conjunto de clases en que se clasificarán los textos. Se denotará por Γ el método de aprendizaje supervisado descrito, el cual actúa como una función que mapea un conjunto de datos de entrenamiento en una función clasificadora, o sea que $\Gamma(D) = \gamma$. La definición ?? implica que cada documento pertenece a una sola clase. Pero existe otro tipo de problemas que permiten que un documento pertenezca a más de una clase. Por ahora este trabajo se enfocará en el tipo de una clase.

Naive Bayes

Feature Selection

K Nearest Neighbor

Medidas de evaluación

Ventajas

Desventajas

Aplicaciones en la Recuperación de la Información

Otros ejemplos de aplicación

Conclusiones

Referencias