Agrupamiento y clasificación en la recuperación de información en la web

Integrantes:
Marcos Manuel Tirador del Riego
Laura Victoria Riera Pérez

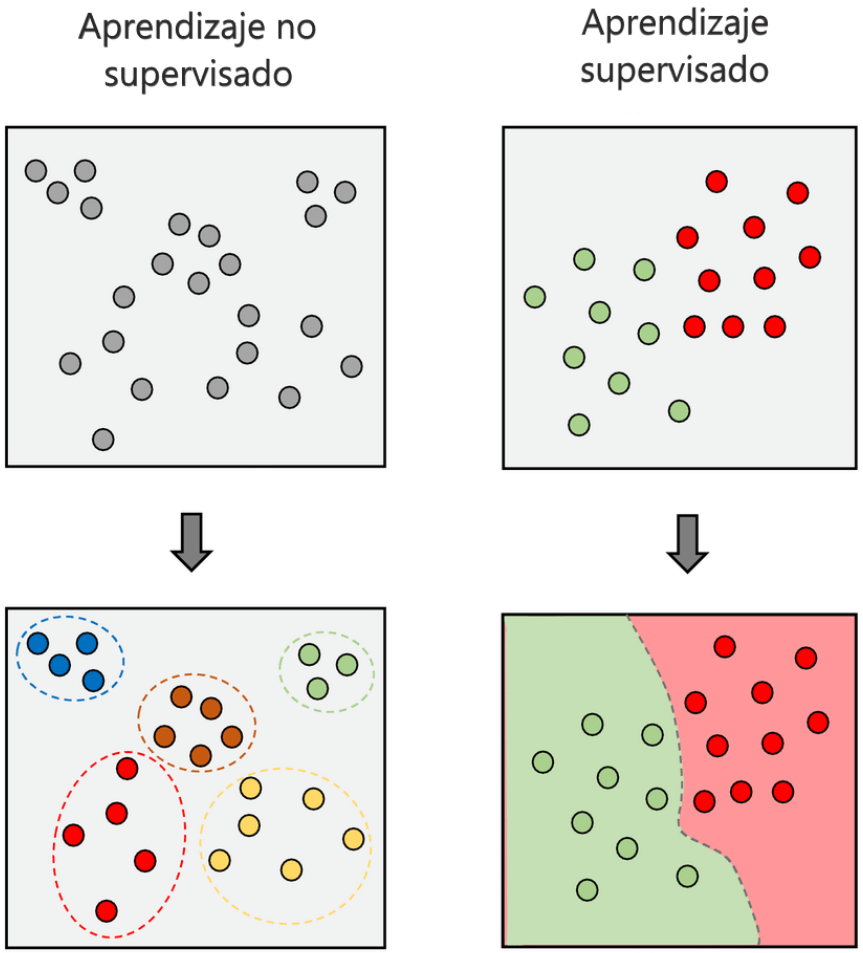
Tercer aÃśo. Ciencias de la computación. Universidad de La Habana. Cuba.

Noviembre, 2022

Índice general I

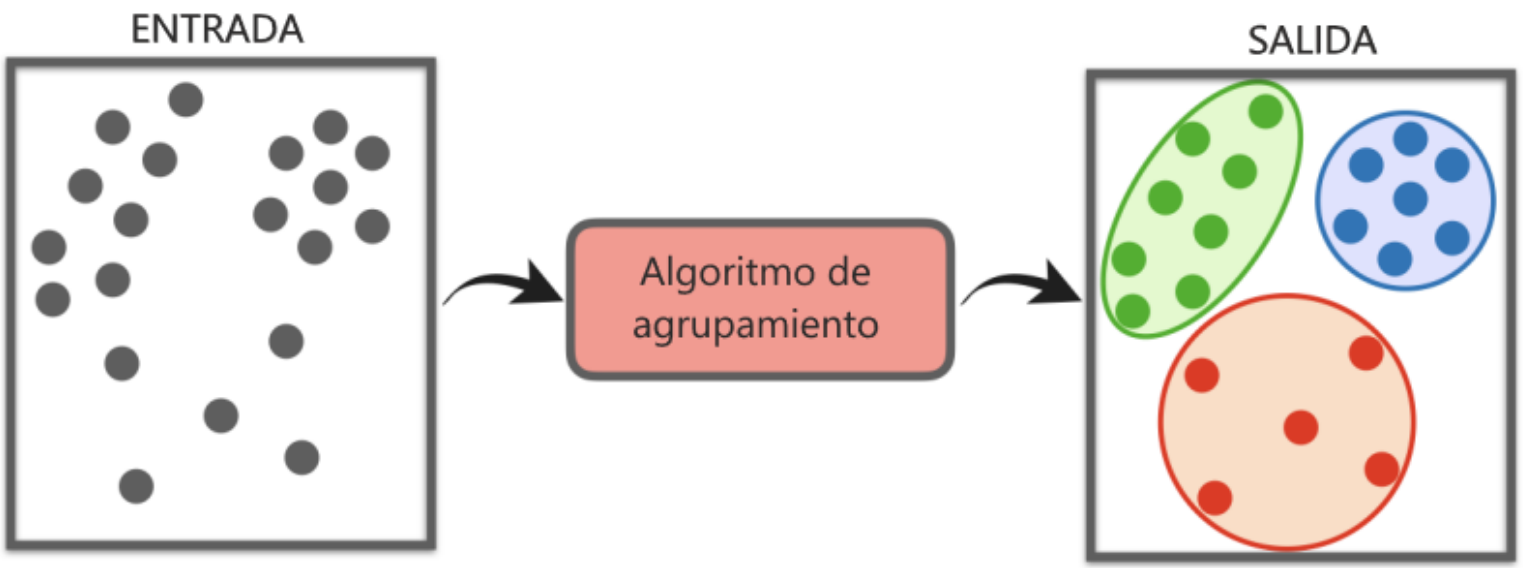
Índice general II

Aprendizaje no supervisado vs. aprendizaje supervisado



Agrupamiento I

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



Características generales

- Es la forma mÃąs comÃžn de aprendizaje no supervisado.
- Los grupos formados deben tener un alto grado de asociaciÃșn 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 informaciÃșn."

La hipÃștesis establece que si hay un documento de un grupo que es relevante a una solicitud de bÃžsqueda, entonces es probable que otros documentos del mismo clÃžster tambiÃ‘n 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 membresÃŋa 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 mÃąs frecuente en el grupo, y luego se mide la precisiÃșn de esta asignaciÃșn contando el nÃžmero de documentos correctamente asignados y dividiendo por N.

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

donde:

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

Se interpreta ω_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 clÃžster si y sÃșlo si son similares. El Ãŋndice de Rand (RI) mide esta precisiÃșn.

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

donde:

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

El Ãŋndice 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 mÃąs fuertemente que los falsos positivos seleccionando un valor $\beta > 1$, dando asÃŋ mÃąs peso al recobrado.

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

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

Agrupamiento particionado

El agrupamiento particionado crea un conjunto de clÃžsteres sin ninguna estructura explÃŋcita que los relacione entre sÃŋ.

K-means I

Es el algoritmo de agrupamiento plano mÃąs 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:

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

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 quÃľ 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 *funciÃșn objetivo* en K-means y nuestro objetivo es minimizarla.

K-means III

Algorithm 1 K-Means

Require: $\{\overrightarrow{x_1}, ..., \overrightarrow{x_N}\}, K$
1: $(\overrightarrow{s_1}, ..., \overrightarrow{s_K}) \leftarrow$ **SelectRandomSeeds**$(\{\overrightarrow{x_1}, ..., \overrightarrow{x_N}\}, K)$
2: **for** $k \leftarrow 1$ **to** K **do**
3: $\quad \overrightarrow{\mu_k} \leftarrow \overrightarrow{s_k}$
4: **while** stopping criterion has not been met **do**
5: $\quad$ **for** $k \leftarrow 1$ **to** K **do**
6: $\quad\quad \omega_k \leftarrow \{\}$
7: $\quad$ **for** $n \leftarrow 1$ **to** N **do**
8: $\quad\quad j \leftarrow \arg\min_{j'} |\overrightarrow{\mu_{j'}} - \overrightarrow{x_n}|$
9: $\quad\quad \mu_j \leftarrow \omega_j \cup \{\overrightarrow{x_n}\}$ (reassignment of vectors)
10: $\quad$ **for** $k \leftarrow 1$ **to** K **do**
11: $\quad\quad \overrightarrow{\mu_k} \leftarrow \frac{1}{|\omega_k|} \sum_{\overrightarrow{x} \in \omega_k} \overrightarrow{x}$ (recomputation of centroids)
12: **return** $\{\overrightarrow{\mu_1}, ..., \overrightarrow{\mu_N}\}$

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 nÃžmero fijo de iteraciones I.
- Cuando la asignaciÃșn 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 jerÃąrquico* produce una jerarquÃŋa, una estructura que es mÃąs informativa que el conjunto no estructurado de clÃžsteres devuelto por el agrupamiento particionado, no requiere que especifiquemos previamente el nÃžmero de grupos y la mayorÃŋa son deterministas, sin embargo son más ineficientes que los particionados. En una representaciÃșn grÃąfica los elementos quedan anidados en jerarquÃŋas con forma de Ãąrbol.
Los algoritmos de agrupamiento jerÃąrquico 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 clÃžster Ãžnico 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 jerÃąrquico aglomerativo o HAC por sus siglas en inglés.
Toman decisiones basadas en patrones locales sin tener inicialmente en cuenta la distribuciÃșn 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 mÃąs 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 mÃąs 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 estÃąn 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 evalÃža la calidad de un clÃžster basada en todas las similitudes entre documentos, evitando asÃŋ castigar valores extremos como en los criterios de enlace Ãžnico 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 clústers está definida como la similitud de sus centroides

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

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 bÃąsico del agrupamiento jerÃąrquico es:

1. Se comienza con N clústeres, resultado de asignar cada elemento al suyo propio. Se computa la matriz C de similitud de NÃůN.
2. Se halla la similitud entre los pares de clústeres con la medida deseada.
3. Se toma el par mÃąs similar de clÃžsteres y se combinan en un Ãžnico clÃžster.
4. Se calculan las similitudes entre el nuevo clÃžster y cada uno de los clústeres antiguos.
5. Se repiten los pasos 3 y 4 hasta que todos los elementos estÃľn agrupados en un solo grupo de tamaÃśo N.

Algoritmo HAC II

Algorithm 2 HAC

Require: $\{d_1, ..., d_N\}$

for $n \leftarrow 1$ **to** N **do**
  for $i \leftarrow 1$ **to** N **do**
    $C[n][i] \leftarrow SIM(d_n, d_i)$
    $I[n] \leftarrow 1$ (keeps track of active clusters)
  $A \leftarrow []$ (assembles clustering as a sequence of merges)
for $k \leftarrow 1$ **to** $N - 1$ **do**
  $< i, m > \leftarrow \arg\max_{<i,m>:i \neq \wedge I[i]=1 \wedge I[m]=1} C[i][m]$
  $A.APPEND(< i, m >)$ (store merge)
  for $j \leftarrow 1$ **to** N **do**
    $C[i][j] \leftarrow SIM(i, m, j)$
    $C[j][i] \leftarrow SIM(i, m, j)$
  $I[m] \leftarrow 0$ (deactivate cluster)
return A

Algoritmo HAC III

Notas:

- En cada iteraciÃșn, los dos clÃžsteres más similares se fusionan y las filas y columnas del clÃžster fusionado i en C se actualizan.
- El agrupamiento se almacena como una lista de fusiones en A.
- I indica quÃ clÃžsteres aÃžn estÃąn disponibles para fusionarse.
- La funciÃșn SIM(i, m, j) calcula la similitud del grupo j con la fusiÃșn de los grupos i y M.

Una suposiciÃșn fundamental en HAC es que la operaciÃșn de fusiÃșn es monótona, es decir si $s_1, s_2, ..., s_{K-1}$ son las similitudes de combinaciÃșn de las fusiones sucesivas de un HAC, entonces se cumple $s_1 \geq s_2 \geq ... \geq s_{K-1}$.

Agrupamiento jerárquico divisivo

Los algoritmos de arriba hacia abajo comienzan con todos los documentos en un grupo. El clÃžster se divide utilizando un algoritmo de agrupamiento particionado. Este procedimiento se aplica recursivamente hasta que cada documento estÃą en su propio clÃžster.
A pesar de necesitar un segundo algoritmo de agrupamiento particionado como una subrutina, tiene la ventaja de ser mÃąs eficiente si no generamos una jerarquÃŋa completa hasta las hojas de documentos individuales. Para un nÃžmero fijo de niveles, y utilizando un algoritmo particionado eficiente como K-means, los algoritmos divisivos son lineales en el nÃžmero de documentos y clÃžsteres.
Además se beneficia de la informaciÃșn completa sobre la distribuciÃșn global al tomar decisiones de particiÃșn 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 Ãžtil 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 aplicaciÃșn en disímiles campos, cuenta el apoyo de una serie de paquetes de software, a menudo disponibles en la informÃątica acadÃľmica 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 tÃľcnica importante para descubrir subregiones o subespacios relativamente densos de una distribuciÃșn de datos multidimensional. Se ha utilizado en la recuperaciÃșn de informaciÃșn para muchos propÃșsitos diferentes, como la expansiÃșn de consultas, la agrupaciÃșn e indexaciÃșn de documentos y la visualizaciÃșn de resultados de bÃžsqueda. Permiten mejorar interfaz y experiencia de usuario y proporcionar una mayor eficacia o eficiencia del sistema de bÃžsqueda.

Ejemplos de aplicación II

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

- Agrupamiento de resultados de bÃžsqueda (Search result clustering): La presentaciÃșn predeterminada de los resultados de bÃžsqueda (documentos devueltos en respuesta a una consulta) en la recuperaciÃșn de informaciÃșn es una lista sencilla. Los usuarios escanean la lista de arriba a abajo hasta que encuentran la informaciÃșn que buscan.
  En su lugar, en la agrupaciÃșn en clÃžsteres de resultados de bÃžsqueda los documentos similares aparecen juntos, siendo mÃąs fÃącil escanear algunos grupos coherentes que muchos documentos individuales. Esto es particularmente Ãștil si un tÃ‍rmino de bÃžsqueda tiene diferentes significados.

Ejemplos de aplicación III

- DispersiÃșn-recopilaciÃșn (Scatter-Gather): Su objetivo es también una mejor interfaz de usuario. Este agrupa toda la colecciÃșn 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 interÃľs.
  La navegaciÃșn basada en la agrupación de clÃžsteres es una alternativa interesante a la bÃžsqueda de palabras clave a la informaciÃșn estÃąndar paradigma de recuperaciÃșn de informaciÃșn. Esto es especialmente cierto en escenarios donde los usuarios prefieren navegar en lugar de buscar porque no estÃąn seguros de quÃľ bÃžsqueda tÃľrminos a utilizar.

Ejemplos de aplicación IV

- Modelado de lenguaje (Language modeling): Explota directamente la hipÃ$tesis del agrupamiento para mejorar los resultados de bÃžsqueda, basado en una agrupaciÃ$n de toda la coleccciÃ$n. Usamos un Ãŋndice invertido estÃąndar para identificar un conjunto inicial de documentos que coincide con la consulta, pero luego agregamos otros documentos de los mismos grupos incluso si tienen poca similitud con la consulta.
  Para evitar problemas de datos escasos en el modelado lenguaje enfocado a RI, el modelo de documento d se puede interpolar con un modelo de coleccciÃ$n. Pero la coleccciÃ$n contiene muchos documentos con tÃľrminos atÃŋpicos de d. Al reemplazar el modelo de coleccciÃ$n con un modelo derivado de grupo de d, obtenemos estimaciones mÃąs precisas de las probabilidades de ocurrencia de tÃľrminos en d.

Ejemplos de aplicación V

- Recuperación basada en clústeres (Cluster-based retrieval): La agrupaciÃșn tambiÃl'n puede acelerar la bÃžsqueda.
  La bÃžsqueda en el modelo de espacio vectorial equivale a encontrar los vecinos mÃąs cercanos a la consulta. El Ãŋndice invertido admite la bÃžsqueda rÃąpida del vecino mÃąs cercano para la configuraciÃșn estÃąndar de RI. Sin embargo, a veces es posible que no podamos usar un Ãŋndice invertido de manera eficiente. En tales casos, podrÃŋamos calcular la similitud de la consulta con cada documento, pero esto es lento. La hipÃștesis de agrupamiento ofrece una alternativa: encontrar los clústeres que estÃąn mÃąs cerca de la consulta y sÃșlo considerar los documentos de estos. Como hay muchos menos clÃžsteres que documentos, se disminuye grandemente el espacio de búsqueda, y encontrar el clÃžster mÃąs cercano es rÃąpido. Además los elementos que coinciden con una consulta son similares entre sÃŋ, por lo que tienden a estar en los

Ejemplos de aplicación VI

mismos clústeres, de esta forma la calidad no disminuye en gran medida.

Clasificación

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