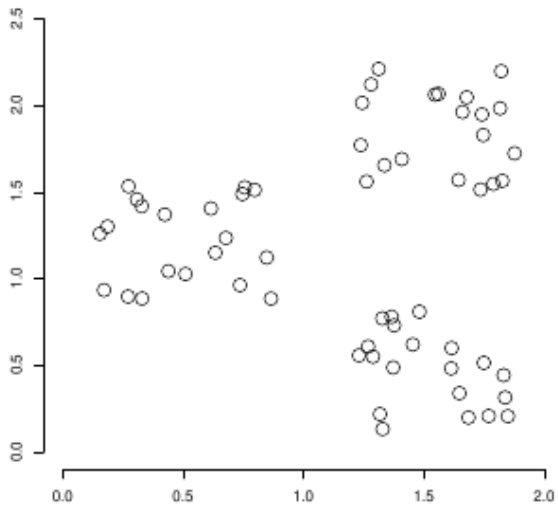

Text clustering

Algoritmos de clustering aplicados a conjuntos de documentos de texto

Marzo de 2020

Alumno (1): David Montero Loaiza
Alumna (2): Javier Alonso del Saso

Índice

Los algoritmos de clustering para documentos de texto permiten agruparlos según su similitud en la información que presentan. Esto permite ordenar el conocimiento de distintos archivos de forma completamente distinta.

Son métodos de aprendizaje no supervisado, por lo que los algoritmos agruparán documentos según el contenido de los mismos. La base fundamental para realizar clústers es definir la distancia entre documentos, generalmente la euclídea.

1. Flat clustering

Flat clustering agrupa a algoritmos de clústers que agrupan los documentos sin ninguna estructura explícita (i.e. los clústers no guardan ninguna relación entre ellos mismos).

Dentro de flat clustering podemos encontrar distintos tipos según clasifiquemos a los documentos:

1. **Hard clustering**: Cuando un documento pertenece a un solo clúster.
2. **Soft clustering**: Cuando un documento pertenece a una distribución de clústers.

1.1. K-means

Tenemos N documentos, $d_i : i = 1..N$, y queremos agrupar los en K grupos, $\omega_i : i = 1..K$ de tal forma que se minimize la función de coste γ. Se dice que tenemos un problema de clustering de N documentos con cardinalidad K.

Definimos un espacio vectorial como el conjunto de todo el vocabulario conocido. Para reconocer a un documento, utilizamos un vector donde cada componente corresponde a un elemento del vocabulario y su valor consistirá en la tf-idf$_{t,d}$.

Ahora, buscamos minimizar la distancia media de los documentos de un clúster al centro del clúster (centroide). Podemos definir dicho centroide como la ponderación de los documentos que lo componen:

$$\vec{\mu}_j = \frac{1}{|\omega_j|} \sum_{\vec{x} \in \omega_j} \vec{x} \tag{1}$$

ω_j es el j-ésimo clúster y los vectores $\vec{x}$ son los documentos contenidos en dicho clúster. $|\omega_j|$ es el número de documentos contenidos en el clúster.

La función de coste se define como las sumas de distancias cuadráticas entre los documentos y

el centroide de un mismo clúster:

$$RSS = \sum_{i}^{K} \left(\sum_{\vec{x} \in \omega_i} |\vec{x} - \vec{\mu}_i|^2 \right)$$

El procedimiento es iterativo, en cada iteración se asigna un nuevo documento a un clúster y se vuelve a calcular el centroide. Esto permite llegar a un mínimo en la función de coste y encontrar agrupamientos. Podemos definir que nos hemos acercado suficiente a los centroides reales cuando la variación de los mismos entre una iteración y otra sea pequeña.

Este algoritmo es sensible a los datos de partida y puede ocurrir que escogamos *outliers* (i.e. documentos muy alejados del resto) y se produzcan clústers de un sólo individuo. Por eso es conveniente repetir el procedimiento y encontrar que se clusters parecidos.

Se puede optimizar el número de clústers óptimos introduciendo un factor λK dentro de RSS y optimizarlo. El factor lambda deberá ser estimado o se pueden aplicar valores frecuentes para encontrar un número de clústers óptimo.

1.1.1. La cardinalidad de los clústers

El método de K-means requiere de antemano que especifiquemos el número de clústers (e.g. la cardinalidad) para clasificar. Esto presenta el problema de cuál es la cantidad que podemos escoger. Se puede decidir coger un número de clústers lo suficientemente alto como para minimizar RSS. Sin embargo, no se conseguiría extraer información de los agrupamientos.

En la práctica, se plantea la siguiente fómula donde tiene en cuenta la complejidad del modelo con un factor λ:

$$K = \arg\min_{K}(RSS_{min}(K) + \lambda K) \tag{2}$$

Donde $RSS_{min}(K)$ es la función de coste para el modelo con cardinalidad K que es mínimo entre varias iteraciones con estados iniciales distintos. Para $\lambda = 0$ no hay restricciones en la complejidad y $\lambda = N$ es un agrupamiento perfecto.

La ventaja que ofrece el determinar λ sobre la cardinalidad del clustering es que se puede emplear en problemas similares (e.g. si quieres clasificar novelas, puedes emplear un lambda fijo que alguien ha usado anteriormente y simplemente minimizas la función anterior).

1.1.2. Algoritmo de maximización de la expectación

Se introduce un algoritmo de *soft clustering*. Permitimos que un documento se pueda encontrar en varios grupos. Permite una mayor flexibilidad a la hora de ordenar los grupos. El modelo de

k-means funciona, en el caso ideal, con distribuciones gausianas radiales donde los centroides corresponden a los centros de estas.

El algoritmo de maximización de la expectación (EM) es un algoritmo iterativo que busca maximizar el *likelyhood* de un conjunto de documentos D dado una serie de parámetros Θ:

$$\Theta = \arg\max_{\Theta} L(D|\Theta) = \arg\max_{\Theta} \sum_{n=1}^{N} \log P(d_n|\Theta) \tag{3}$$

Donde d_n es un documento.

El algoritmo EM usa una distribución *prior* sobre los clústers. Se asume que sigue una distribución de Bernoulli donde $q_{mk} = P(U_m = 1|\omega_k)$ es la probabilidad de que un documento dentro del clúster ω_k contenga el elemento del vocabulario (i.e. el término), t_m. Utilizando el teorema de Bayes podemos razonar que la probabilidad tiene la forma:

$$P(d|\Theta) = \sum_k \alpha_k \left(\prod_{t_m \in d} q_{mk} \right) \left(\prod_{t_m \notin d} (1 - q_{mk}) \right) \tag{4}$$

Donde α_k es la probabilidad de que un documento se encuentre dentro del clúster ω_k sin condiciones. El conjunto de parámetros Θ_k para un clúster ω_k son $\alpha_k, q_{1k}, ..., q_{Mk}$, donde M es el número total de elementos del vocabulario.

De la misma forma que en el algoritmo de k-means, existe una función de maximización (de la función *likelihood*) para aproximar los parámetros, y otra función para establecer la distribución de un documento entre los clústers.

En k-means, los centroides se actualizaban hayando la media de todos los documentos dentro del clúster. Ahora, buscamos α_k y q_{mk}:

$$q_{mk} = \frac{\sum_n r_{nk} I_{mn}}{\sum_n r_{nk}} \tag{5}$$

Encontramos que para hayar la probabilidad, ponderamos por r_{nk} que es la probabilidad de que el documento n se encuentre en el clúster ω_k. El término I_{mn} recoge sólo a los documentos que contienen el término en cuestión.

$$\alpha_k = \frac{\sum_n r_{nk}}{N} \tag{6}$$

El factor α se pondera sobre la cantidad de documentos.

Ahora, r_{nk} se calcula usando los parámetros anteriormente definidos:

$$r_{nk} = \frac{\alpha_k P(d|\omega_k;\Theta)}{P(d|\Theta)} \tag{7}$$

Donde $P(d|\omega_k;\Theta)$ es la probabilidad de encontrar el documento d en el clúster ω_k. Uno de los términos de la ecuación (4).

1.2. Métodos de verificación

Existen varios estimadores para obtener un valor cuantitativo de la calidad de la clasificación de los documentos. Será necesario conocer las clases a las que pertenecen los documentos. Esto se puede conseguir a través de inspección humana y determinar previamente a que clase corresponde.

Sin embargo, existen varias formas dependiendo del problema. Por ejemplo, en el caso de búsquedas de documentos usando una query, podemos tomar el tiempo que tarda el usuario en seleccionar el documento deseado como una valoración de la calidad de la búsqueda.

1.2.1. Pureza

Podemos medir el número de documentos pertenecientes a la clase mayoritaria dentro de cada clúster y ponderar por el número de documentos. Nos ofrece una idea de cuánto de bien agrupan los clústers a las clases. Sin embargo, no tiene en cuenta el número de clústers y siempre se pueden añadir más hasta clasificar una clase por clúster.

1.2.2. Normalized Mutual Information

Cuantificación de la información que ofrecen los clústers. Se normaliza usando la entropía (complejidad) del sistema de clústers y de clases.

$$\mathrm{NMI} = \frac{\mathrm{I}}{(H_{\mathrm{cluster}} + H_{\mathrm{class}})/2} \tag{8}$$

Donde I representa la información que ofrecen los clústers sobre las clases;

$$\mathrm{I} = \sum_{k=1}^{K}\sum_{i=1}^{L} P(\omega_k \cap c_i) \log \frac{P(\omega_k \cap c_i)}{P(\omega_k)\ P(c_i)} \tag{9}$$

Y las H's son las entropías;

$$H(\mathbf{x}) = -\sum_i P(x_i)\log(x_i) \tag{10}$$

Tiene rango entre 0 (las agrupaciones aleatorias) a 1 (los clústers representan perfectamente a las clases). Al estar normalizado por la entropía de los clusters, no se puede mejorar los agrupamientos creando subdivisiones.

1.2.3. Rand Index

El rand index (RI) maneja el porcentaje de pares de documentos que han sido clasificados como similares (en el mismo clúster) o desimilares (en clústers distintos).

Existen un total de $N(N-1)/2$ de pares de documentos. Para calcular el RI necesitamos conocer los casos True Positive (TP), True Negative (TN), False Positive (FP) y False Negative (FN). Entendemos como los casos positivos como aquellos que se han clasificado como similares y se encuentran dentro del mismo clúster.

El total de decisiones positivas;

$$\mathrm{TP} + \mathrm{FP} = \sum_{k=1}^{K} \binom{|\omega_k|}{2} \tag{11}$$

Donde $|\omega_k|$ es el tamaño del clúster. Se asume que no existen singletones (clústers con un sólo documento).

Para calcular los casos verdaderos (usando información adicional para asignar clases);

$$\mathrm{TP} = \sum_{\omega_k} \sum_{c_i \in \omega_k} \binom{|c_i|}{2} \tag{12}$$

Con esta información podemos despejar para FP y de la misma forma aplicarlo a los casos negativos.

El IR se define como;

$$\mathrm{IR} = \frac{TP + TN}{TP + FP + TN + FN} \tag{13}$$

El estadístico F utiliza la misma fórmula pero ponderando los casos negativos mucho más con un factor β ¿1:

$$P = \frac{\mathrm{T}P}{\mathrm{T}P + \mathrm{F}P} \quad R = \frac{\mathrm{T}P}{\mathrm{T}P + \mathrm{F}N} \quad F(\beta) = \frac{(\beta^2 + 1)PR}{\beta^2 P + R}$$

1.3. Aplicaciones

Flat clustering nos permite conseguir nueva información que anteriormente no se podía comprender con un grupo de documentos. Existen varias utilidades, según se quiera ofrecer información a un cliente u organizar un grupo de archivos en disco.

1.3.1. Resultados de búsqueda

Podemos proporcionar los resultados de búsqueda con query como clústers. Esto funciona muy bien cuando una palabra tiene varios significados. Por ejemplo, buscando el término banco nos pueden aparecer dos clústers, uno sobre el asiento público y otro sobre el tipo de empresa.

1.3.2. Scatter-Gather

Esta aplicación consiste en reorganizar un conjunto de documentos por clústers. El usuario interactúa con dichos clústers, seleccionando aquellos de interés. Luego, se vuelven a generar nuevos clústers con los documentos que se encontraban en los grupos anteriores. Este método de búsqueda permite sustituir la query por la navegación sobre agrupamientos, mucho más intuitivo.

1.3.3. Eficiencia en las búsquedas

Como calcular la similitud de una query con un documento es costoso, se puede calcular la similitud de un clúster con una query (e.g. utilizando el centroide de dicho clúster) y asignando los documentos del clúster a los resultados de la búsqueda. Esto permite mucha mayor eficiencia en los motores de búsqueda.

Por otra parte, podemos realizar el proceso inverso, encontrar los documentos más similares al query y devolver los documentos que se encuentren dentro del mismo clúster. Esto permite devolver documentos que tengan términos que no coincidan con el query pero que la información ofrecida sea similar.

2. Clustering Jerárquico

El flat clustering es un método simple y eficaz para la clusterización de textos y documentos, no obstante, hace falta conocer la cantidad de clusters en los cuales se quiere categorizar la información, lo que lleva a la decisión de cuántos clústers deben utilizarse para realizar esta agrupación, y cuyo cálculo no es determinístico.

El clustering jerárquico, a diferencia del flat clustering, nos devuelve como resultado una jerarquía de textos o documentos que se encuentran asociados a una estructura, lo cual le da un valor adicional al agrupamiento realizado.

Existen dos tipos de clustering jerárquico: el clustering jerárquico aglomerativo y el clustering jerárquico divisivo. El clustering jerárquico aglomerativo se basa en una dirección de subida, de manera que inicia con simples documentos que finalmente se agrupan en un solo cluster; y el clustering jerárquico divisivo, que a diferencia del aglomerativo, tiene una dirección de bajada, de manera que inicia en la parte superior y se irá dividiendo en pequeños pares de clusters hasta llegar a los simples documentos.

2.1. Clustering Jerárquico Aglomerativo

El clustering jerárquico aglomerativo (HAC) es un método por el cual se agrupan los documentos desde el fondo hasta la cima, es decir, se consideran inicialmente cada uno de los documentos como simples nodos, o clusters, que se irán uniendo por pares a otros documentos, generando así diversos clusters, y cuya unión irá creciendo hasta haber unido todos los documentos en un solo cluster.

Los HAC son visualizados por medio de dendogramas, en donde cada una de las uniones son representadas por medio de líneas horizontales, cuya longitud (coordenada y) representa la combinación de similitud del cluster que une cada par de documentos. Esta combinación mantiene una propiedad de monotonicidad, sobre la cual se asume que para una cantidad n de combinaciones de similitudes sucesivas s_i, s_i siempre será mayor o igual que s_{i+1}.

Los HAC no necesitan de un número predefinido de clústers para ser generado, no obstante, particiones del dendograma pueden ser realizadas para obtener clústers necesarios en algún tipo de análisis. Para realizar la partición, el dendograma debe ser cortado uniformemente en su coordenada y, para lo cual existen diversas formas de hacerlo.

1. Cortar el dendograma teniendo en cuenta un valor predefinido de similaridad. Si queremos tener clusters con una combinación de similitud mínima, por lo tanto cortamos el dendograma por ese valor de combinación de similitud mínimo.

2. Cortar el dendograma donde exista el valor más alto de combinación de similitud, que representa una gran brecha en el dendograma. Esta brecha generalmente representa una clusterización natural de los documentos analizados.

3. Cortar el dendograma donde la suma residual de los cuadrados más una regularización sea la mínima. En este caso, la fórmula sería la siguiente:

$$K = argmin[RSS(K') + \lambda K']$$

   Donde K es el corte asociado a una cantidad K de clústers, K' es el corte asociado a una cantidad K' de clústers, RSS es la suma residual de cuadrados y λ es un parámetro de

penalización.

Para aumentar la versatilidad de este método, cualquier medida de distorción, además del RSS, puede ser usada en este caso.

4. Cortar el dendograma por un número de clústers predefinido, seleccionando el punto de corte que genera esta cantidad de clústers.

Para calcular un HAC, es necesario calcular la matriz de similitud C, la cual tendrá un tamaño $N \times N$. Como método iterativo, los clústers más similares se unen y la matriz C se actualiza con los nuevos valores para este cluster.

Las medidas de similitud pueden variar, existiendo cuatro tipos de clustering: clustering por unión simple, clustering por unión completa, clustering por media de grupos y clustering por centroides.

1. **Clustering por enlace simple.** En el clustering por enlace simple, la similitud entre clústers viene dada por la similitud de sus documentos más similares. Este tipo de clustering no tiene en cuenta la estructura entera del clúster, lo que puede llevar al caso de extender una cadena de puntos con una combinación de similitud muy bajas; a esto se le llama el efecto cadena.

2. **Clustering por enlace completo.** En el clustering por enlace completo, la similitud entre clústers viene dada por la similitud de sus documentos más distantes. Este tipo de clustering tiene en cuenta la estructura entera del clúster, pero al utilizar los documentos más distantes, es susceptible a presentar errores en el caso de que existan documentos atípicos en uno o en ambos clústers.

3. **Clustering por media de grupos.** El clustering por media de grupos (GAAC) denota la calidad de un clúster teniendo en cuenta todas las similitudes entre documentos, por lo tanto, teniendo en cuenta la estructura entera de los clústers a evaluar y evitando así los problemas generados por los clustering por enlance simple y por enlace completo.

   La similitud se calcula por medio de la medida $SIM - GA$, que viene dada por:

   $$SIM - GA(w_i, w_j) = \frac{1}{(N_i+N_j)(N_i+N_j-1)}[(\sum_{d_m \in w_i \cup w_j} \vec{d_m})^2 - (N_i + N_j)]$$

   Donde w_i y w_j son los clústers i y j respectivamente, N_i y N_j son la cantidad de documentos en cada clúster y $\vec{d}$ es el vector cuya longitud ha sido normalizada del documento d.

4. **Clustering por centroides.** El clustering por centroides tiene en cuenta, al igual que el GAAC, la estructura entera de los clústers a evaluar, no obstante, la similitud se calcula

por medio de la medida $SIM - CENT$, que viene dada por:

$$SIM - CENT(w_i, w_j) = \frac{1}{N_i N_j} \sum_{d_m \in w_i} \sum_{d_n \in w_j} \vec{d}_m \cdot \vec{d}_n$$

No obstante, teniendo en cuenta esta ecuación, el clustering por centroides, a diferencia del GAAC, no tiene en cuenta pares de documentos del mismo clúster.

Cabe resaltar, que teniendo en cuenta el clústering por centroides, la propiedad de monoticidad puede romperse, y en algunos casos, la similitud de uniones sucesivas puede aumentar en vez de disminuir.

2.2. Clustering Jerárquico Divisivo

El clustering jerárquico divisivo es un método por el cual se desagrupan los documentos desde la cima hasta el fondo, en donde quedarán cada uno de los documentos. Este tipo de clustering jerárquico inicia con un clúster que contiene todos los documentos, el cual se va dividiendo en pares de clústers utilizando un algoritmo de flat clustering recursivamente hasta que cada documento queda en un solo clúster.

A pesar de que el clustering jerárquico divisivo requiere de otro algoritmo adicional basado en flat clustering, tiene una ventaja sobre el HAC cuando no se desean realizar particiones hasta que cada documento sea un clúster por sí mismo, ya que puede detenerse el algoritmo en la cantidad de clústers deseados y por lo tanto aumentando la eficiencia de este.

3. Etiquetado de clústers

El etiquetado de clústers es la parte final de la clusterización, tanto en flat clustering como en clustering jerárquico, ya que, usualmente, el ser humano es aquel que interactúa con los clústers de documentos y por ende necesita saber qué diferencia a cada clúster.

Existen dos enfoques para realizar el etiquetado de clústers: el etiquetado de clústers diferencial y el etiquetado de clúster interno.

1. **Etiquetado de clústers diferencial.** El etiquetado de clústers diferencial selecciona etiquetas para cada clúster teniendo en cuenta la comparación de la distribución de los términos en un clúster contra los demás.

2. **Etiquetado de clústers interno.** El etiquetado de clústers interno, a diferencia del etiquetado de clústers diferencial, no tiene en cuenta la distribuación de términos de otros

clústers. Uno de los métodos más usados en este tipo de etiquetado es el de usar los títulos de los documentos cercanos al centroide del clúster.

En el caso de la clusterización jerárquica, el etiquetado de clústers se vuelve un poco más complejo debido a la propiedad padre/hijos. Los clústers hijos tienen una distribución de términos que, aunque sea diferente, terminará siendo la misma para el clúster padre, que finalmente tendrá que lidiar con la distribución de términos de sus clústers hijos, de manera que puede darse el caso en que las etiquetas asignadas a los clústers no diferencien adecuadamente los temas de estos.

4. Últimos avances

Cuando se trata de una gran colección de documentos (i.e. $< 100,000$) es importante tener en cuenta la eficiencia de los métodos algorítmicos. En [DFY01] se ha empleado la computación en paralelo para generar el espacio vectorial del vocabulario de todos los documentos en un tiempo relativamente corto (23 min). Se tuvieron en cuenta los tiempos de lectura de documentos en disco así como de usar tablas hash que tienen reducido espacio, sabemos que los datos en el espacio vectorial están muy esparcidos.

La polirepresentación se ha empleado usando técnicas de clustering [AF15] para optimizar la búsqueda de documentos. La polirepresentación consiste en usar distintas representaciones de la información (e.g. documentos, resúmenes de artículos, revisiones, comentarios, títulos) que optimizan la búsqueda al usuario. Se pueden aplicar solapamientos entre las representaciones para obtener los artículos más relevantes según una gran variedad de información. Para obtener las representaciones se puede aplicar métodos de clustering lo que permiten resultados de búsquedas mucho más optimizados.

En 2013 se aplicó algoritmos de clustering para mejorar los resultados de búsquedas utilizando querys [RK13]. Mediante *Markov Random Fields*, se realiza un ranking de los clusters. Se aprovecha el uso de grafos que se construyen con los clusters y la query para obtener valores de similitud entre llos y producir un mejor ranking que el inicial y que permita enseñar al usuario documentos que son más relevantes según la búsqueda.

Referencias

[AF15] Muhammad Kamran Abbasi and Ingo Frommholz. Cluster-based polyrepresentation as science modelling approach for information retrieval. *Scientometrics*, 102(3):2301–2322, 2015.

[AOO+07] Ismail Sengör Altingövde, Rifat Ozcan, Huseyin Cagdas Ocalan, Fazli Can, and

Özgür Ulusoy. Large-scale cluster-based retrieval experiments on Turkish texts. In *Proc. SIGIR*, pages 891–892. ACM Press, 2007.

[DFY01] Inderjit S. Dhillon, James Fan, and Guan Yuqiang. Efficient clustering of very large document collections. In Robert et al. [RKK+01], pages 357–381.

[MRS09] Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze. *An Introduction to Information Retrieval*. Cambridge University Press, 2009.

[RK13] Fiana Raiber and Oren Kurland. Ranking document clusters using markov random fields. In *Proceedings of the 36th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '13, page 333–342, New York, NY, USA, 2013. Association for Computing Machinery.

[RKK+01] L. Grossman Robert, Chandrika Kamath, Philip Kegelmeyer, Vipin Kumar, and Raju R. Namburu, editors. *Data Mining for Scientific and Engineering Applications*, volume 2. Springer, Boston, MA, 2001.