Algoritmo Kmeans

El algoritmo Kmeans es un algoritmo iterativo que intenta dividir el conjunto de datos en subgrupos (clústeres) distintos y no superpuestos definidos previamente por K donde cada punto de datos pertenece a un solo grupo. Intenta hacer que los puntos de datos intra-clúster sean lo más similares posible y al mismo tiempo mantiene los clústeres lo más diferentes (lejos) posible. Asigna puntos de datos a un grupo de modo que la suma de la distancia al cuadrado entre los puntos de datos y el centroide del grupo (media aritmética de todos los puntos de datos que pertenecen a ese grupo) es mínima. Cuanta menos variación tengamos dentro de los grupos, más homogéneos (similares) serán los puntos de datos dentro del mismo grupo.

La forma en que funciona el algoritmo kmeans es la siguiente:

1. Especifique el número de grupos K.
2. Inicialice los centroides barajando primero el conjunto de datos y luego seleccionando aleatoriamente K puntos de datos para los centroides sin reemplazo.
3. Siga iterando hasta que no haya cambios en los centroides. es decir, la asignación de puntos de datos a los clústeres no está cambiando.

- Calcule la suma de la distancia al cuadrado entre los puntos de datos y todos los centroides.
- Asigne cada punto de datos al grupo más cercano (centroide).
- Calcule los centroides de los grupos tomando el promedio de todos los puntos de datos que pertenecen a cada grupo.

El enfoque que sigue kmeans para resolver el problema se llama Expectativa-Maximización. El paso E consiste en asignar los puntos de datos al grupo más cercano. El paso M es calcular el centroide de cada grupo.

La función objetivo es:

$$J = \sum_{i=1}^{m} \sum_{k=1}^{K} w_{ik} \|x^i - \mu_k\|^2$$

donde

w_{ik} = 1 para el punto de datos xi si pertenece al grupo k; de lo contrario, w_{ik}= 0. Además, u_k es el centroide del grupo de x^i.

Es un problema de minimización de dos partes. Primero minimizamos J w.r.t. wik y tratar μk fijo. Entonces minimizamos J. u_k y tratar w_{ik} arreglado. Técnicamente hablando, primero derivamos J con respecto de w_{ik} y actualice las asignaciones de clústeres (E-pas). Luego derivamos J con respecto de u_k y vuelva a calcular los centroides después de las asignaciones de clúster del paso anterior (M-paso). Por lo tanto, E-paso es:

$$\frac{\partial J}{\partial w_{ik}} = \sum_{i=1}^{m} \sum_{k=1}^{K} \|x^i - \mu_k\|^2$$

$$\Rightarrow w_{ik} = \begin{cases} 1 & Si \;\; k = argmin_j \|x^i - \mu_j\|^2 \\ 0 & En\ otro\ caso \end{cases}$$

En otras palabras, asigne el punto de datos x^i. al grupo más cercano juzgado por su suma de la distancia al cuadrado del centroide del grupo.

Y M-paso es:

$$\frac{\partial J}{\partial \mu_k} = 2 \sum_{i=1}^{m} w_{ik}(x^i - \mu_k) = 0$$

$$\Rightarrow \mu_k = \frac{\sum_{i=1}^{m} w_{ik} x^i}{\sum_{i=1}^{m} w_{ik}}$$

Lo que se significa en volver a calcular el centroide de cada grupo para reflejar las nuevas asignaciones.

Algunas cosas a tener en cuenta son:

- Dado que los algoritmos de agrupación en clústeres que incluyen kmeans utilizan mediciones basadas en la distancia para determinar la similitud entre los puntos de datos, se recomienda estandarizar los datos para que tengan una media de cero y una desviación estándar de uno, ya que casi siempre las características de cualquier conjunto de datos tendrían diferentes unidades de medida. como la edad frente a los ingresos.
- Dada la naturaleza iterativa de kmeans y la inicialización aleatoria de los centroides al comienzo del algoritmo, diferentes inicializaciones pueden llevar a diferentes clústeres, ya que el algoritmo de kmeans puede quedarse atascado en un óptimo local y no converger al óptimo global. Por lo tanto, se recomienda ejecutar el algoritmo utilizando diferentes inicializaciones de centroides y seleccionar los resultados de la ejecución que arrojaron la suma más baja de la distancia al cuadrado.
- La asignación de ejemplos no cambia es lo mismo que ningún cambio en la variación dentro del clúster:

$$\frac{1}{m_k} \sum_{i=1}^{m_k} \|x^i - \mu_{c^k}\|^2$$

Los métodos de inicialización de Forgy y Partición Aleatoria son comúnmente utilizados. El método Forgy elige aleatoriamente k observaciones del conjunto de datos y las utiliza como centroides iniciales. El método de partición aleatoria primero asigna aleatoriamente un clúster para cada observación y después procede a la etapa de actualización, por lo tanto calcular el clúster inicial para ser el centro de gravedad de los puntos de la agrupación asignados al azar. El método Forgy tiende a dispersar los centroides iniciales, mientras que la partición aleatoria ubica los centroides cerca del centro del conjunto de datos. Según Hamerly y compañía, el método de partición aleatoria general, es preferible para los algoritmos tales como los k-medias armonizadas y fuzzy k-medias. Para expectation maximization y el algoritmo estándar el método de Forgy es preferible.

Ventajas:

- Sencillo
- Rápido.

Desventajas:

- Es necesario decidir el valor de *k.*
- El resultado final depende de la inicialización de los centroides.
- En principio no converge al mínimo global sino a un mínimo local.

Complejidad

- Respecto a la complejidad computacional, el agrupamiento k-medias para problemas en espacios de d dimensiones es:
- NP-hard en un espacio euclidiano general d incluso para 2 grupos 1112
- NP-hard para un número general de grupos k incluso en el plano 13
- Si k y d son fijados, el problema se puede resolver en un tiempo $O(n^{dk+1} * log(n))$ donde n es el número de entidades a particionar