K-Means

Nicolás Kossacoff

Octubre 2024

1. Introducción

K-means es un método estadístico que busca clasificar observaciones en grupos (i.e., clusters) mutuamente excluyentes. Su objetivo final es que al comparar observaciones dentro de un mismo cluster, las observaciones sean muy parecidas entre sí, pero al comparar esas mismas observaciones con las observaciones de otros clusters sean distintas.

Se definimos a nuestro conjunto de datos como $D = \{X_1, \ldots, X_n\}$ y definimos $\mathcal{C} = \{C_1, \ldots, C_K\}$ como una partición de D, entonces cada sub-conjunto de la partición $\mathcal{C}$ es un **cluster**.

1.1. Medida de similaridad

En cada cluster queremos tener observaciones que sean lo más similares posibles. Para eso, definimos la **dispersión intra-grupo** como:

$$W(C_k) = \frac{1}{2N} \sum_{j=1}^{n} \sum_{i=1}^{n} ||X_i - X_j||_2^2 \tag{1}$$

la cual calcula la distancia promedio entre las observaciones de un cluster. Nos da una idea de que tan parecidas son las observaciones dentro de cada cluster.

Podemos reescribir la Ecuación (1) como la suma de las distancias entre las observaciones y el centroide del cluster. Entonces:

$$W(C_k) = \frac{1}{2N} \sum_{j=1}^{n} \sum_{i=1}^{n} ||X_i - X_j||_2^2 = \sum_{i=1}^{n} ||X_i - \bar{X}_k||_2^2 \tag{2}$$

donde $\bar{X}_k$ es un vector de largo p con la media de cada una de las p features.

Finalmente, definimos la **dispersión total intra-grupos** como:

$$W(\mathcal{C}) = \sum_{k=1}^{K} W(C_k) \tag{3}$$

1.1.1. Definición de distancia

Como pudimos observar en la Sección 1.1, cómo calculamos la distancia entre las observaciones es muy importante. Esto va a terminar definiendo qué significa que dos observaciones sean similares entre sí, lo que termina por definir la forma y tamaño de nuestros clusters.

Entonces, la medida de distancia óptima es la que mejor se adapte al problema en cuestión. Por ejemplo, en las anteriores ecuaciones utilizamos la distancia Euclídea, la cual es una buena elección si nuestros datos siguen una distribución aproximadamente normal. Ahora, si nuestros datos no estuviesen normalmente distribuidos, o hubiesen outliers en nuestra muestra, utilizar la distancia Euclídea no es la mejor opción, y en ese caso otras medidas como la distancia Manhattan, Minkowski o Coseno son preferidas[1].

2. Algoritmo

La idea original de K-means es la de encontrar la partición $\mathcal{C}$ que minimiza la dispersión total intra-grupo:

$$\min_{C_1,\ldots,C_k} \sum_{k=1}^{K} W\left(C_k\right) \tag{4}$$

Tal como se puede apreciar en la Figura 1, cuanto más compactos sean los clusters (i.e., cuanto más cerca del centroide estén las observaciones) mejor.

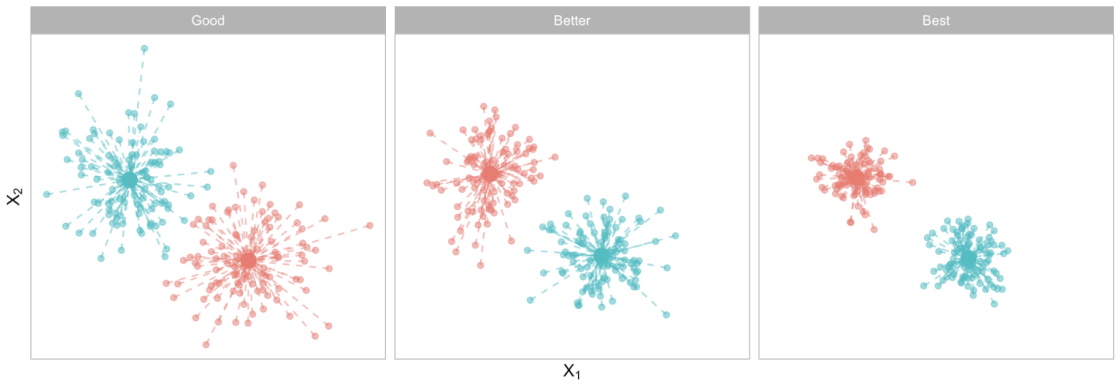

Figura 1: Diferentes soluciones del algoritmo K-means según la distancia intra-grupo que se alcanza.

Ahora, el problema es que tendríamos que tomar todas las posibles particiones y calcular la distancias intra-grupo para cada uno de los clusters que las componen, lo cual lo convierte en un problema computacionalmente muy costoso.

[1]Detalle a tener en cuenta: si la medida de distancia que utilizamos no es la distancia Euclídea, entonces el método estadístico no se llama K-Means sino K-Medoids, el cual es una generalización del primero.

Para entender mejor este problema, notar que la cantidad de formas en las que se pueden agrupar n observaciones en K clusters depende de cuantas particiones no vacías se pueden obtener para un mismo conjunto (i.e., hay aproximadamente K^n posibles particiones), lo que a su vez está relacionado con los números de Stirling. Ese número de posibles particiones crece exponencialmente con la cantidad de observaciones, n, y con la cantidad de features, k. Por lo tanto, incluso con valores chicos de n y de k, el costo computacional es muy alto.

2.1. Solución local

Debido al costo computacional que implica calcular todos los posibles clusters, el algoritmo de K-means se concentra en **soluciones locales**.

Comienza eligiendo, aleatoriamente, K observaciones para utilizar como centroides de los clusters iniciales. Luego, para cada observación, calcula su distancia con cada uno de los centroides, y la asigna al cluster con el centroide más cercano. Una vez construidos los nuevos clusters, vuelve a calcular los centroides y las distancias, asignando nuevamente las observaciones a los clusters con los centroides más cercanos. Este procedimiento se repite hasta que la solución converge, es decir, hasta que los clusters no cambian de un paso a otro.

Debido a la aleatoriedad al elegir las K observaciones iniciales, los resultados que obtenemos al realizar este procedimiento pueden variar significativamente. Es por eso que es recomendable realizar varias iteraciones de este procedimiento y finalmente quedarnos con los clusters que más reduzcan la dispersión total intra-grupo, $W(C_k)$.

2.1.1. Resumen

El algoritmo se puede resumir de la siguiente manera:

1. Elegimos la cantidad de clusters, K.
2. Elegimos aleatoriamente K observaciones, las cuales utilizamos como los centroides de los clusters iniciales, $\{C_1, \ldots, C_K\}$.
3. Para cada observación $i \in \{1, \ldots, n\}$, calculamos la distancia respecto a cada uno de los centroides, y la asignamos al cluster con el centroide más cercano. Es decir:
$$i \in C_l \text{ si } ||X_i - \bar{X}_l||_2^2 = \arg \min_{1 \leq k \leq K} ||X_i - \bar{X}_k||_2^2 \tag{5}$$
En este paso obtenemos los nuevos clusters, $\mathcal{C}' = \{C'_1, \ldots, C'_K\}$.
4. Calculamos los centroides de los nuevos clusters, $\{\bar{X}_1, \ldots, \bar{X}_K\}$, y la dispersión total intra-grupo, $W(\mathcal{C}')$.
5. Repetimos los pasos (3) y (4) hasta que el algoritmo converge, es decir, los resultados no cambian de un paso a otro.