Evaluación de un Método Mejorado del Algoritmo K-Means Aplicado en un Sistema de Recuperación de Documentos

Osvaldo Mario Spositto[1] [0000-0002-7472-0938], Julio César Bossero[1] [0000-0002-2498-9103], Sebastián Quevedo[1] [0009-0008-0249-3685], Viviana Alejandra Ledesma[1] [0000-0003-4218-2474], Lorena Romina Matteo[1] [0000-0002-5873-5793]

[1] Departamento de Ingeniería e Investigaciones Tecnológicas.
Universidad Nacional de La Matanza.
Florencio Varela 1903. San Justo. La Matanza. Buenos Aires. Argentina
{spositto,jbossero,jquevedo,vledesma,lmatteo}@unlam.edu.ar

Abstract. Este trabajo presenta una evaluación exhaustiva de un método mejorado del algoritmo K-Means, aplicado a un sistema de recuperación de documentos. Los algoritmos de agrupamiento, o clustering, se utilizan para organizar documentos similares en grupos, basándose en características comunes como el contenido textual, la frecuencia de palabras y otros atributos relevantes. El método mejorado evaluado en este estudio introduce optimizaciones que reducen significativamente el tiempo de procesamiento necesario para asignar cada documento a un clúster. Esto se logra mediante una combinación de técnicas de preprocesamiento y ajustes en los criterios de convergencia del algoritmo, resultando en una menor cantidad de iteraciones y operaciones computacionales. Para evaluar el método mejorado, se llevaron a cabo una serie de experimentos utilizando un prototipo propio que construye artificialmente un conjunto de datos de documentos. En la fase de experimentación, se aplicaron tanto el algoritmo K-Means estándar como el método mejorado estudiado. La comparación se realizó en términos de tiempo de procesamiento, número de iteraciones y precisión en la agrupación de documentos. Para evaluar la calidad del agrupamiento, se empleó el coeficiente de silueta o silhouette como métrica. Los resultados mostraron que, además de reducir el tiempo de procesamiento, el método mejorado mantiene una calidad de agrupamiento equivalente a la obtenida con el K-Means clásico. Esto sugiere que las optimizaciones introducidas no comprometen la precisión del agrupamiento, sino que, por el contrario, ofrecen un rendimiento más eficiente sin sacrificar la efectividad en la clasificación de documentos.

Keywords: Agrupamiento; K-Means; Inicialización; Coeficiente de Silueta.

1 Introducción

En los *Sistemas de Recuperación de Información* (SRI), también conocidos como *Sistemas de Recuperación de Documentos* (SRD), la incorporación de algoritmos de agrupamiento o clustering, se comenzó a utilizar para agrupar documentos similares en

conjuntos, lo que le permite al sistema presentar los resultados más relevantes al usuario. En lugar de mostrar una lista larga y desorganizada de documentos, el sistema puede ofrecer agrupaciones de temas relacionados, facilitando la navegación. Estos han captado gran interés entre investigadores académicos y científicos [1]. Estos métodos se han estudiado también recientemente debido a la aplicabilidad en otras áreas tales como motores de búsqueda, minería web, etc. [2]. Al organizar los documentos en grupos o clústeres, se minimiza la necesidad de comparar una consulta con cada documento de forma individual. En su lugar, el sistema puede enfocar la búsqueda en los grupos más relevantes, lo que optimiza el proceso de recuperación ya que se reduce el número de comparaciones necesarias. El uso de esas estas técnicas proponen brindar una estructura ordenada al agrupar documentos similares en conjuntos coherentes. De este modo, se espera minimizar el 'ruido' en los resultados de búsqueda y mejorar significativamente la experiencia del usuario. Así, la búsqueda de información podría volverse más efectiva y enfocarse en los temas de interés del usuario, facilitando la localización de datos relevantes.

Este grupo de investigación presentó distintos trabajos acerca de la implementación en SRI de distintos algoritmos de aprendizaje de máquina, para clasificar documentos automáticamente en grupos temáticos similares [3-5].

Con el objetivo de lograr una mayor discriminación de objetos en grupos homogéneos, se estudiaron diversas técnicas de clustering o agrupamiento, pertenecientes al aprendizaje no supervisado. Según la bibliografía consultada, K-Means se destaca como la técnica de clustering por particiones más utilizada [6].

Según Ahmed y otros [7], este algoritmo es ampliamente reconocido como uno de los métodos de minería de datos más potentes y populares entre los investigadores. No obstante, pese a su popularidad, presenta ciertas limitaciones, como los problemas derivados de la inicialización aleatoria de los centroides, que pueden resultar en una convergencia inesperada. Además, continúan los autores, expresando que este tipo de algoritmo requiere que se especifique el número de clústeres de antemano, lo cual influye en las diferentes formas que pueden adoptar los clústeres y en cómo se manejan los valores atípicos. Otra limitación del algoritmo es su incapacidad para manejar diversos tipos de datos y el alto costo que consume muchos recursos computacionales y de tiempo de ejecución en la detección de grupos de modo automático.

Este documento, tomando una parte del trabajo de Duracik [8], realiza un análisis experimental de una modificación al algoritmo K-Means. En la implementación presentada, se introduce una mejora clave en el proceso de asignación de vectores a clústeres. Tradicionalmente, en el algoritmo, cada vector se asigna al clúster cuyo centroide está más cercano. Sin embargo, esta versión modificada añade una condición adicional para la reasignación, lo cual ayuda a mejorar la estabilidad del algoritmo y acelerar su convergencia. La mejora se encuentra en el paso de reasignación de vectores. En lugar de simplemente mover un vector al clúster más cercano, se compara la distancia del vector al centroide actual con el doble de la distancia al centroide más cercano en una lista de clústeres candidatos. Específicamente, un vector solo se reasignará si la nueva distancia es menor de la distancia al clúster actual. Esta condición adicional evita que los vectores oscilen innecesariamente entre clústeres cercanos, lo que puede ocurrir debido a pequeñas diferencias en las distancias durante las iteraciones. Al evitar estas reasignaciones innecesarias, el algoritmo no solo mejora su eficiencia, reduciendo el número de cambios en cada iteración, sino que también

estabiliza los resultados finales. Esta estabilidad es importante para asegurar que el algoritmo converja a una solución óptima o cercana a la óptima sin iteraciones excesivas. En resumen, esta mejora optimiza el proceso de clustering al hacer que el algoritmo sea más robusto frente a pequeñas variaciones en las distancias y facilita una convergencia más rápida y confiable.

La organización del documento es como sigue: la Sección 1 introduce los conceptos básicos que explican de manera general el contexto y el propósito del trabajo; la Sección 2 presenta el análisis de los trabajos relacionados, uno de los cuales se toma como base en la experimentación; en la sección 3 se describen las distintas técnicas utilizadas. En la sección 4 se detalla el diseño del experimento. En la sección 5 se muestran y analizan las conclusiones y las líneas de trabajos futuros. Finalmente, en la sección 6 se hacen los agradecimientos.

2. Trabajos Relacionados

Como ya se mencionó este trabajo se basa, en parte, en la propuesta que se titula "Método optimizado basado en algoritmo K-Means como herramienta en la detección de plagio de código fuente" [8], en este artículo se presenta un método mejorado del algoritmo K-Means, el cual permite dividir las líneas de código en grupos y así ahorrar tiempo y operaciones innecesarias en la búsqueda de coincidencias.

La publicación de Pérez-Ortega et al. [9], trata sobre una mejora al algoritmo K-Means, la cual está orientada a la solución eficiente de instancias con un gran número de grupos y de dimensiones. Esta heurística está basada en la relación entre el número de dimensiones y el número de centroides que conforman una vecindad, permitiendo reducir el número de cálculos de distancias objeto-centroide para cada objeto.

En los siguientes apartados se hace referencias a otras modificaciones que se conocen del algoritmo K-Means clásico:

K-Medoids: es una variante del algoritmo K-Means que se utiliza para el clustering de datos. A diferencia de K-Means, que utiliza la media de los puntos en un clúster como centroide, K-Medoids selecciona puntos de datos reales como los centros de los clústeres. Esto lo hace más robusto a valores atípicos y a datos con distribuciones no gaussianas [10].

K-Means++: mejora la inicialización de los centroides para reducir la probabilidad de convergencia a soluciones subóptimas [11].

Otros trabajos presentan estudios comparativos de estas propuestas, con los que, se orientan futuros esfuerzos de investigación.

K-Medians: usa la mediana en lugar de la media para minimizar la suma de las distancias absolutas, siendo más robusto a valores atípicos [12].

3. Marco Teórico

Para realizar una experimentación, que permita comparar la calidad de agrupamiento, entre el algoritmo K-Means tradicional y el método mejorado, se

desarrolló un prototipo[1] simulador en lenguaje C#. El aplicativo permite configurar parámetros, generando un modelo vectorial. El modelo se basa en matrices que representan la relación entre términos y documentos, cada posición de la matriz (*i, j*), representa el valor de la frecuencia con la que el término *j* aparece en el documento *i.*

Para determinar el número óptimo de grupos o clústeres (*k*) en un conjunto de datos, se utilizó el algoritmo del Codo (Elbow Method). El método evalúa la suma de las distancias cuadradas dentro del clúster en función del número de clústeres.

La evaluación comparativa entre los métodos se realizó empleando el coeficiente de Silueta (Silhouette), también conocida como puntuación de la silueta (Silhouette Score), que sirve como una métrica para evaluar la calidad de los grupos formados por métodos de agrupación. El propósito de esta métrica es determinar la similitud dentro de cada grupo y contrastarla con la similitud presente en otros grupos.

3.1 El Algoritmo K-Means

El algoritmo K-Means, también conocido como agrupamiento duro (hard clustering), es un método que asigna cada dato a un único clúster de manera exclusiva. Esto significa que no se permiten pertenencias parciales ni ambigüedades: cada punto de datos se asocia completamente a un clúster específico según un criterio de similitud o distancia [13]. Así, los datos se dividen en grupos distintos, con cada elemento perteneciendo exactamente a uno de ellos.

Uno de los principales desafíos computacionales del método K-Means es identificar características que permitan agrupar elementos similares. El objetivo de este método es particionar un conjunto de n observaciones en k grupos, de manera que cada observación se asigne al grupo cuyo centroide esté más cercano, minimizando así la distancia promedio dentro de cada grupo. Para utilizar el algoritmo K-Means, primero se especifica el número de clústeres deseados (k). Por ejemplo, si se establece "k" en 3, habrá 3 clústeres. Cada clúster está representado por un centroide, que es la media de los puntos de datos asignados a él Figura 1.

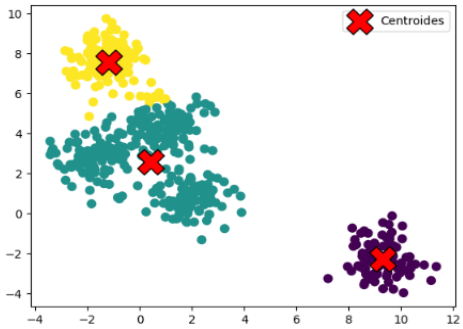

Figura 1. Representación gráfica usando Google Colab para un ejemplo de K-Means con 3 grupos, en código en Python. Este ejemplo utiliza la biblioteca scikit-learn para realizar el agrupamiento y matplotlib para visualizar los resultados. Elaboración propia.

[1] El prototipo se describió en un trabajo presentado y aprobado en el XXVI Workshop de Investigadores en Ciencias de la Computación (WICC 2024), realizado los días 18 y 19 de abril de 2024, en la ciudad de Puerto Madryn. Chubut.

El algoritmo funciona de manera iterativa, ajustando las asignaciones de los puntos hasta que cada uno esté más cerca de su propio centroide que de los de otros clústeres, minimizando así la distancia interna en cada iteración. A continuación, se detalla cómo funciona el algoritmo K-Means paso a paso en un seudocódigo:

```
1: function K_Means(V : vector list, k : int, C : list
   of clusters): list of clusters
2:     repeat
3:         change ← 0
4:         for all vector v in V do
5:             c_old ← LastCluster(v)
6:             d_min ← Distance(v, c_old)
7:             for all cluster c in C do
8:                 d ← Distance(v, c)
9:                 if d < d_min then
10:                    d_min ← d
11:                    c_old ← c
12:                end if
13:            end for
14:            if LastCluster(v) != c_old then
15:                Assign(v, c_old)
16:                change ← change + 1
17:            end if
18:        end for
19:        for all cluster c in C do
20:            Recalculate(c)
21:        end for
22:    until change = 0
23:    return C
24: end function
```

No obstante, la aleatoriedad de los valores de k puede ocasionar diferentes formas de agrupación. Por ello, es necesario definir adecuadamente su valor con el objetivo de agrupar a solo los datos que contengan la mayor cantidad de atributos similares [11].

3.2 Método del Codo

El método Elbow o Codo ayuda a elegir el número óptimo de clústeres, cuando se busca hacer clasificación en un conjunto de datos.

Este método muestra la relación entre la varianza y la cantidad de clústeres de manera gráfica, representando el "codo" el punto en el que agregar más clústeres no explica significativamente más varianza, lo que nos permite seleccionar una cantidad óptima de clústeres que represente bien los datos sin que se produzca un sobreajuste. Para la identificación del "codo", es decir para determinar el número óptimo de clústeres [14], se utiliza el indicador de la suma de cuadrados dentro del clúster (WCSS por sus siglas en inglés Within-Cluster Sum of Squares,), en función del número de clústeres, es decir, el punto donde la velocidad de disminución de WCSS se ralentiza repentinamente.

Para hacer uso de este método se parte del cálculo de la distorsión promedio de cada clúster, esto es la distancia de cada elemento con su centroide correspondiente. Para calcular la distorsión se usa:

$$\boldsymbol{distorsión} = \frac{\sum_{i=1}^{N} || \boldsymbol{x_i - centroide} ||^2}{\boldsymbol{N}} \quad \textbf{[1]}$$

donde:

N es el número de elementos.

x_i es el i-ésimo punto de datos.

centroide es el centroide del clúster al que pertenece el punto x_i.

$//x_i - centroide//^2$ es la distancia euclidiana cuadrada entre el punto x_i y su centroide.

La distorsión mide la compacidad de los clústeres: cuanto menor sea la distorsión, más cerca estarán los puntos de sus centroides respectivos, indicando una mejor agrupación. En el contexto del método del codo, al graficar la distorsión contra el número de clústeres *k*, buscamos el punto donde la tasa de disminución de la distorsión se reduce notablemente. Este punto, conocido como el "codo", nos ayuda a determinar el número óptimo de clústeres.

3.3 Coeficiente de Silueta:

Este coeficiente es una métrica para evaluar la calidad del agrupamiento obtenido con algoritmos de clustering.

El coeficiente de Silueta [15] para una observación *i* se denota como *s(i)* y se define como:

$$\text{Silueta} = \mathbf{1/N} \sum_{n=1}^{N} \left(\frac{b_i - a_i}{\max \{b_i , a_i\}} \right) \quad \textbf{[2]}$$

El valor de la silueta es una medida de cuán similar es un objeto a su propio grupo o clúster (cohesión) en comparación con otros grupos (separación). La silueta va de -1 a +1, donde: un valor positivo indica que el objeto está bien emparejado con su propio grupo o clúster, si es 0 la observación *i* está entre dos grupos y si el valor es negativo significa que está mal asignada a su grupo o clúster. La métrica puede ser calculada con cualquier fórmula de distancia, como la distancia Euclidiana o la distancia Manhattan.

4. Evaluación Experimental

En esta experimentación, se utilizó un prototipo propio desarrollado para generar matrices que representen un Modelo Vectorial (MV), con el propósito de evaluar la calidad de los métodos de agrupamiento. En el MV, tanto los documentos como las consultas en lenguaje natural se representan mediante vectores de términos, que ocupan un espacio lineal multidimensional. Este modelo se basa en la similitud entre una consulta dada por el usuario y los documentos de una colección, cuyos términos han sido ponderados utilizando la técnica TF-IDF (Term Frequency – Inverse Document Frequency). Esta técnica mide la relevancia de un término para un documento dentro de un corpus (o conjunto de documentos), considerando tanto la frecuencia de aparición del término en el documento como su presencia en el resto de la colección. TF-IDF es una medida numérica que se utiliza frecuentemente como factor de ponderación en los

SRI, ya que permite evaluar la importancia relativa de cada término en un documento en relación con la consulta del usuario[16][17].

El simulador permite a través de distintas configuraciones crear artificialmente una matriz que represente un corpus voluminoso, y representarlo como una matriz de términos/documentos, siendo los términos las columnas que representan las orientaciones temáticas, y las filas los diferentes documentos. Esto se conoce como el MV. En la Figura 2 se observa un ejemplo de una matriz resultante.

	0	1	2	3	4	5	6
0	0.915625	-0.108594	0.339063	0.010937	0.334375	-0.14375	0.615625
1	0.090625	0.629687	0.015625	0.7375	-0.185937	0.669531	0.0625
2	0.45625	0.601562	0.221875	0.6625	-0.017188	0.282813	0.424023
3	0.355469	0.882812	0.596875	-0.195312	0.001563	0.992969	-0.069922
4	-0.171875	0.463281	-0.125	0.676562	0.60625	0.458594	0.75625
5	0.2875	-0.141406	0.7375	0.303906	0.826563	0.953125	0.690625
6	0.957812	0.414062	-0.146094	0.746875	0.076563	-0.0125	0.054297
7	0.484375	-0.171875	0.095312	0.871094	0.385938	0.615625	0.139844

Figura 2. Matriz documento-término. Elaboración propia

Esta herramienta es altamente configurable, permitiendo ajustar la cantidad de dimensiones de la matriz y la cantidad de tuplas, lo que ofrece una flexibilidad para adaptarse a diferentes escenarios y necesidades de investigación.

Otra característica, es la capacidad de probar con distintas métricas de distancias, lo que enriquece la evaluación y análisis de los agrupamientos. En la aplicación del algoritmo en este trabajo, se utilizó la distancia euclidiana como medida de similitud. Estas medidas determinan cómo se evalúa la proximidad entre los datos y se define como la longitud de la línea recta que conecta dos puntos en el espacio euclidiano, calculada mediante el teorema de Pitágoras. La distancia euclidiana es una medida directa de la distancia más corta entre dos puntos y cumple con las propiedades de una métrica, como la no negatividad, la simetría y la desigualdad triangular [18]. En la siguiente figura se muestra el resultado de una ejecución del prototipo utilizado para evaluación y seguimiento de agrupamientos ante las modificaciones incorporadas.

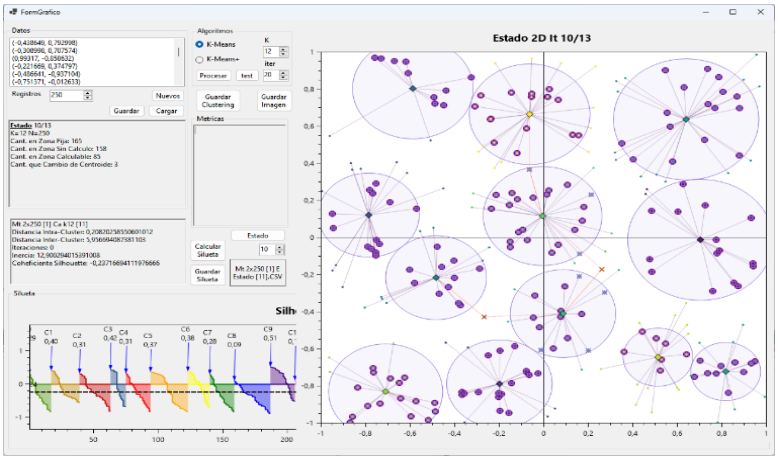

Figura 3. Pantalla donde se visualizan métricas y resultados.

5. Conclusión y Líneas de Trabajos Futuros

En las pruebas realizadas, se analizaron los espacios dimensionales utilizando un asistente gráfico bidimensional para comprender el fundamento de la optimización y el proceso de cada iteración. Los resultados confirmaron que la cantidad de cálculos necesarios en la versión optimizada es siempre menor, y que esta cantidad disminuye a medida que los centroides dejan de desplazarse. Se determinó que la cantidad de cálculos está regida por una fórmula que varía para cada punto del clúster, en función de su posición relativa respecto al centroide.

$$cd_t = \begin{cases} 0, & \leftrightarrow t = 0 \\ k \times N, & \leftrightarrow t = 1 \\ \binom{k}{2} + k + \sum_{i=1}^{k} \left(\sum_{j=1}^{n_i} \begin{cases} 0, & \leftrightarrow j \in PS_i \\ 1, & \leftrightarrow j \in (PF_i - PS_i) \\ c(CP_{i,j}) + 1, & \leftrightarrow j \in (P_i - PF_i) \end{cases} \right), & \leftrightarrow t > 1 \end{cases} \quad \textbf{[3]}$$

donde:

- t es el número de iteración.
- cd_t es la cantidad de cálculos de distancia en la iteración t.
- k es la cantidad de clústeres.
- n_i es la cantidad de puntos del clúster i.
- $c(x)$ es la cantidad de elementos en el conjunto.
- P_i es el conjunto de puntos correspondientes al clúster i al inicio de la iteración.
- PS_i es el conjunto de puntos que no recalculan distancia en el clúster i.
- PF_i es el conjunto de puntos dentro de la zona fija del clúster i.
- $CP_{i,j}$ es el conjunto de centroides dentro de la zona de cálculo del punto j respecto del centroide del clúster i.

Se ha determinado que las optimizaciones no pueden implementarse en la primera iteración, ya que en esta etapa los puntos aún no están asignados a ningún clúster. Posteriormente, se deben considerar: primero, la distancia entre los distintos centroides, que se calcula mediante la combinatoria de k en 2, y segundo, la suma de k para calcular el desplazamiento de los centroides desde la iteración anterior.

Los conjuntos de puntos y centroides mencionados en la fórmula 3 aparecen representados en la Figura 4, que grafica en 2 dimensiones este análisis en la décima iteración del ejemplo. Allí los centroides se representan como diamantes y los puntos del clúster (conjunto P_i) como puntos unidos por líneas hacia el centroide de igual color. La mencionada zona fija aparece como un círculo morado traslucido con radio igual a la mitad de distancia al centroide más cercano. Dentro de ella están los puntos que ya sabemos que permanecerán en el clúster al final de la iteración aún sin calcular su distancia hacia ningún otro centroide (conjunto PF_i). Entre ellos hay puntos redondeados en morado, para los cuales ni siquiera necesitaremos calcular su distancia actual al centroide (conjunto PS_i), puesto que esa distancia en la iteración anterior sumada a la distancia del desplazamiento del centroide sigue siendo inferior al radio de la zona fija. Esta variación del algoritmo nos permitió omitir cada vez más cálculos a medida que los centroides minimizan su desplazamiento, hasta quedar inmóviles.

Por último, todos los puntos fuera de la zona fija del propio clúster han de tener un tratamiento diferente al del K-Means original. En la figura se ve un punto j redondeado en rojo. A su alrededor un círculo rojo con radio igual a la distancia del punto j al

centroide *i*. Cualquier centroide que esté dentro, en definitiva, está más cerca del punto que su actual centroide. El círculo naranja tiene radio igual al doble de esa distancia, y abarca a los centroides con los que se calculará la distancia y comparará (conjunto $CP_{i,j}$). Aunque abarca varios centroides más que el círculo rojo, igualmente sirve para reducir el conjunto de posibilidades respecto del total.

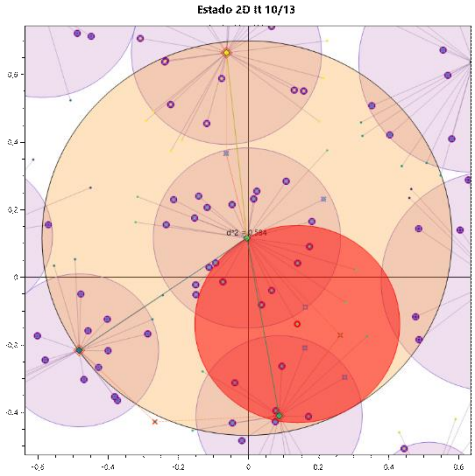

Figura 4. Pantalla del prototipo donde se visualizan puntos y áreas distinguidas.

En la Tabla 1 se ve la comparativa entre cálculos y proporción para un ejemplo para el mismo conjunto de centroides iniciales generados al azar (K-Means original). En la misma se ve cómo se llega al mismo resultado en iteraciones, calidad de agrupamiento (evaluado por silueta y detallado en Figura 5) e incluso en la definición de los clústeres, pero con una cantidad de cálculos de distancia notablemente inferior.

Tabla 1. Comparación entre el K-Means original y el optimizado.

Medida	Original	Optimización	Comparación
Silueta	-0.24	-0.24	100%
Iteraciones	13	13	100%
Cantidad Cálculos	39000	7303	18,72%

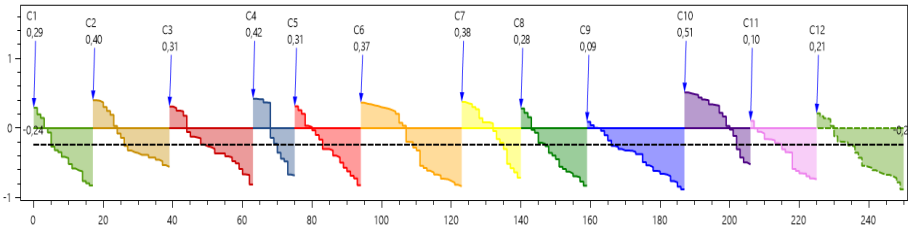

Figura 5. Resultados de la prueba para K-Means.

En trabajos futuros, se pretende investigar la posible relación entre la cantidad de cálculos ahorrados (en comparación con el algoritmo K-Means original) y la métrica

de silueta u otras métricas similares. Este análisis podría revelar si las iteraciones que optimizan el agrupamiento también resultan en un menor costo computacional.

Además, se explorará una línea de trabajo que busca utilizar estas optimizaciones para reducir aún más los cálculos de distancia entre los puntos y los clústeres. Esto se logrará tomando en cuenta el radio del clúster (la máxima distancia desde el centroide a un punto) y una parametrización que considere los cambios en la iteración anterior, así como la precisión deseada en el agrupamiento.

6. Agradecimientos

Este trabajo se realiza en el contexto del Programa de Incentivos a Docentes Investigadores de la Secretaría de Políticas Universitarias (ProInce) que corresponde al período (2023-2024). Se agradece al Departamento de Ingeniería e Investigaciones Tecnológicas (DIIT) de la Universidad Nacional de La Matanza por el apoyo a la línea de investigación del trabajo que se titula “Estudio del proceso de recuperación de información en corpus voluminosos mediante una modificación del algoritmo K-Means para universos de gran dimensión y múltiples segmentos”.

Referencias

1 Ordoñez Quintero, Cristian & Ordonez, Armando & Méndez, Cristian & Ordoñez, Hugo. (2020). Evaluación e implementación de técnicas de clustering para un sistema de recuperación de documentos judiciales. RISTI - Revista Ibérica de Sistemas e Tecnologías de Información.pp. 141-151.

2 Bordignon, F., & Tolosa Chacón, G., (2007), “Recuperación de información: un área de investigación en crecimiento. Ciencias de la Información, 38(1-2),13-24, ISSN: 0864-4659-2

3 Ryckeboer H y otros (2018), Uso de Minería de Datos para Acelerar la recuperación de documentos. CACIC 2018. ISBN: 978-950-658-472-6.

4 Ryckeboer H. y otros. (2018), Recuperación de información acelerada con Algoritmos de Minería de Datos. ECEFI 2018. ISBN: 978-950-42-0191-5.

5 Spositto O. y otros (2020), Hacia la Optimización de un Sistema de Recuperación de Información, WICC 2020. ISBN 978-987-3714-82-5.

6 Núñez Barrionuevo, O. F. (2020). Agrupación por características de consumo eléctrico de parroquias de la provincia de Pichincha Ecuador mediante el algoritmo K-Means.

7 Ahmed, M., Seraj, R., & Islam, S. M. S. (2020). The K-Means algorithm: A comprehensive survey and performance evaluation. Electronics, 9(8), 1295.

8 Duracik, M., Callejas-Cuervo, M., & Mikusova, M. (2020). Método optimizado basado en algoritmo K-Means como herramienta en la detección de plagio de código fuente. Revista Ibérica de Sistemas e Tecnologias de Informação, (E29), 620-632.

9 Pérez-Ortega, Joaquín y otros. (2018). Una heurística eficiente aplicada al algoritmo K-means para el agrupamiento de grandes instancias altamente agrupadas. Computación y Sistemas, 22(2), 607-619. Epub 21 de enero de 2021.https://doi.org/10.13053/cys-22-2-2546

10 Reddy, B. O., & Ussenaiah, D. M. (2012). Literature survey on clustering techniques. IOSR Journal of Computer Engineering, 3(1), 1-50.

11 Bahmani, B., Moseley, B., Vattani, A., Kumar, R., & Vassilvitskii, S. (2012). Scalable k-means++. arXiv preprint arXiv:1203.6402.

12 Moshkovitz, M., Dasgupta, S., Rashtchian, C., & Frost, N. (2020, November). Explainable k-means and k-medians clustering. In International conference on machine learning (pp. 7055-7065). PMLR.
13 Rai, P., & Singh, S. (2010). A survey of clustering techniques. International Journal of Computer Applications, 7(12), 1-5.
14 Liu, F., & Deng, Y. (2020). Determine the number of unknown targets in open world based on elbow method. IEEE Transactions on Fuzzy Systems, 29(5), 986-995.
15 Banchero, Santiago. (2015) Calidad del agrupamiento: Coeficiente de Silueta. Bases de Datos Masivas. Dpto de Cs. Básicas. USAM. Disponible en: https://www.labredes.unlu.edu.ar/sites/www.labredes.unlu.edu.ar/files/site/data/bdm/coeficiente-silueta.pdf.
16 Kowalski G. /1997). Information Retrieval Systems: Theory and Implementation, 1st ed. Norwell, MA, USA: Kluwer Academic Publishers.
17 Salton, G., Wong, A., & Yang, C. S. A, (1975), Vector Space Model for Information Retrieval. Communications of the ACM, 18(11), 613–620, Disponible en: https://doi.org/10.1145/361219.361220.
18 Torres Ricart, S., Alonso Díaz, D., Martínez Sánchez, N., & Merced Len, S. (2023). Uso del algoritmo K-Means para clasificar ciudadanos cubanos mediante un cuestionario de estilos de vida. Disponible en: https://rein.umcc.cu/bitstream/handle/123456789/2354/2.%20Sheyla%20Torres%20Ricart.pdf?sequence=1&isAllowed=y