# A greedy set cover approach for the recomputation of the cluster centres in the FPAC Algorithm

Ivan Feliciano[1] and Edgar Hernandez-Gonzalez[2]

[1] National Institute of Astrophysics, Optics and Electronics
ivan.felavel@gmail.com
[2] National Institute of Astrophysics, Optics and Electronics
edgarmoy.28@gmail.com

**Abstract.** En este trabajo presentamos una modificación al algoritmo K-means usando una heurística que permite hacer el recalculo de centroides de una manera diferente al k-means tradicional. El procedimiento se basa en... primero, después, por ultimo.

**Keywords:** First keyword · Second keyword · Another keyword.

## 1 Introduction

- El número de documentos que se producen hoy en día
- Escribir el problema de agrupamiento
- Escribir para qué sirve el agrupamiento
- Describir el algoritmo Kmeans

Actualmente el numero de documentos en la web aumenta rápidamente, por tal motivo se necesitan algoritmos capaces de agrupar automáticamente grandes cantidades de datos. K-means es un algoritmo de agrupamiento, su objetivo es particionar un conjunto de datos en k grupos basándose en sus características. El agrupamiento se realiza minimizando la suma de distancias entre cada objeto y el centroide de su grupo [1]. El algoritmo consta de tres pasos: 1. Inicialización: una vez escogido el número de grupos, k, se establecen k centroides en el espacio de los datos, por ejemplo, escogiéndolos aleatoriamente. 2. Asignación objetos a los centroides: cada objeto de los datos es asignado a su centroide más cercano. 3. Actualización centroides: se actualiza la posición del centroide de cada grupo tomando como nuevo centroide la posición del promedio de los objetos pertenecientes a dicho grupo. Se repiten los pasos 2 y 3 hasta que los centroides no se mueven, o se mueven por debajo de una distancia umbral en cada paso. A pesar de que el algoritmo K-means es muy popular no es escalable para datos de gran tamaño y dimensión. El principal cuello de botella de K-means es asignar cada vector no centroide a un grupo.

## 2 Related Work

- Describir los intentos hasta lo que hizo el autor
- Describir el algoritmo del autor
- Describir el problema con el algoritmo del autor

Se han hecho varias aportaciones para mejorar el algoritmo k-means. [la del articulo] desarrollo un algoritmo de partición rápida basado en una heurística de los vecinos mas cercanos. Dado un conjunto de centroides, evitar el calculo de distancia por pares entre vectores para obtener una partición de la colección, en su lugar se ocupó una asignación basada en el vecino mas cercano de cada centro, para esto se utilizo una lista invertida de vectores dispersos. También se evito el costoso calculo del centroide verdadero de cada grupo, se propuso una heurística para elegir el centroide de manera eficiente. En [9] los autores utilizan la heurística más lejana primero que consiste en seleccionar los centroides iniciales y evitar los cálculos de distancia redundante desde los no centroides a los centros. En [12, 19 y 20] se utilizaron estructuras de datos de partición de espacio como kd-trees. Esto aumenta la eficacia de k-means, pero solo para pocas dimensiones En [23] reasignaciones de clúster ocurren frecuentemente para puntos que no están cerca de los centroides. Identifica estos puntos al agrupar puntos vecinos usando múltiples arboles de partición. En [2 y 18] se utiliza una heurística la cual elije aleatoriamente el primer centroide y utiliza una distribución de probabilidad. En [4] se utilizo k-means escalable en el cual cada centroide se ve como una consulta para recuperar una lista de documentos que luego se asignan a ese grupo sin cálculo de distancia. [2] k-means ++.[2]

## 3 Proposed Solution

- Describir en el paso en el algoritmo en que nos concentramos
- Describir la adaptación que se hizo a los demás pasos para que utilizara varios centroides, initCentroids() y getClosestCluster()
- Tal vez sea bueno escribir un poco sobre el problema de ser cover y la aproximación greedy
- introducir un poco de lo que se habla en la siguiente subsección

### 3.1 Recomputation of the cluster centres

- Describir el procedimiento
- Tal vez la complejidad y desventajas

## 4 Experiments

### 4.1 Dataset

qué conjuntos de datos utilizamos

- Por qué no usamos el TREC Microblog Dataset
- 20 news, descripción del dataset
- sentiment140 tweets
- gender tweets

### 4.2 Clustering evaluation

- Medidas que utilizamos para evaluar el clustering y hablar un poco de las clases que hay de cada conjunto que se utilizó
- Purity
- NMI
- RI

### 4.3 Implementation

Decir que utilizamos como base lo desarrollado por Ganguly usando Lucene

### 4.4 Compared approaches

Decir que comparamos esta versión del FPAC M centroids con K means normal, SKMeans y FPAC.

### 4.5 Parameters

- El valor de K
- el número de iteraciones
- El valor de M

## 5 Results

- Por cada dataset una gráfica del NMI, Purity y RI
-

## 6 Conclusions and future work

## References

1. Ganguly, D.: A fast partitional clustering algorithm based on nearest neighbours heuristics. Pattern Recognition Letters **112**, 198–204 (2018). https://doi.org/10.1016/j.patrec.2018.07.017
2. Manning, C.D., Raghavan, P., Schütze, H.: Introduction to information retrieval. Cambridge University Press (2009)