EDUCACIÓN
SECRETARÍA DE EDUCACIÓN PÚBLICA

TECNOLÓGICO NACIONAL DE MÉXICO

Tecnológico Nacional de México

Centro Nacional de Investigación y Desarrollo Tecnológico

Tesis de Maestría

Caracterización de dos mejoras del estado del arte del algoritmo *K-Means* orientadas a la solución de grandes instancias

presentado por

Ing. Carlos Fernando Moreno Calderón

como requisito para la obtención del grado de

Maestro en Ciencias de la Computación

Director de tesis

Dr. Joaquín Pérez Ortega

Codirectora de tesis

Dra. María Yazmin Hernández Pérez

Cuernavaca, Morelos, México. Enero de 2022

cenidet
Centro Nacional de Investigación y Desarrollo Tecnológico

Centro Nacional de Investigación y Desarrollo Tecnológico
Departamento de Ciencias Computacionales

Cuernavaca, Mor., 17/enero/2022

OFICIO No. DCC/007/2022
Asunto: Aceptación de documento de tesis
CENIDET-AC-004-M14-OFICIO

DR. CARLOS MANUEL ASTORGA ZARAGOZA
SUBDIRECTOR ACADÉMICO
PRESENTE

Por este conducto, los integrantes de Comité Tutorial del C. CARLOS FERNANDO MORENO CALDERÓN, con número de control M19CE059, de la Maestría en Ciencias de la Computación, le informamos que hemos revisado el trabajo de tesis de grado titulado **"CARACTERIZACIÓN DE DOS MEJORAS DEL ESTADO DEL ARTE DEL ALGORITMO K-MEANS ORIENTADAS A LA SOLUCIÓN DE GRANDES INSTANCIAS"**, y hemos encontrado que se han atendido todas las observaciones que se le indicaron, por lo que hemos acordado aceptar el documento de tesis y le solicitamos la autorización de impresión definitiva.

DR. JOAQUÍN PÉREZ ORTEGA
Director de tesis

DRA. MARÍA YASMÍN HERNÁNDEZ PÉREZ
Codirector de Tesis

DRA. ALICIA MARTÍNEZ REBOLLAR
Revisor 1

DR. JAVIER ORTIZ HERNÁNDEZ
Revisor 2

Revisor 3

C.c.p. Depto. Servicios Escolares.
Expediente / Estudiante
JGGS/ibm

Cuernavaca, Mor., 06/junio/2022
No. De Oficio: SAC/78/2022
Asunto: Autorización de impresión de tesis

CARLOS FERNANDO MORENO CALDERÓN
CANDIDATO AL GRADO DE MAESTRO EN CIENCIAS DE LA COMPUTACIÓN
P R E S E N T E

Por este conducto, tengo el agrado de comunicarle que el Comité Tutorial asignado a su trabajo de tesis titulado **"Caracterización De Dos Mejoras Del Estado Del Arte Del Algoritmo K-Means Orientadas A La Solución De Grandes Instancias"**, ha informado a esta Subdirección Académica, que están de acuerdo con el trabajo presentado. Por lo anterior, se le autoriza a que proceda con la impresión definitiva de su trabajo de tesis.

Esperando que el logro del mismo sea acorde con sus aspiraciones profesionales, reciba un cordial saludo.

A T E N T A M E N T E
Excelencia en Educación Tecnológica®
"Educación Tecnológica al Servicio de México"

EDUCACIÓN | TECNOLÓGICO NACIONAL DE MÉXICO
CENTRO NACIONAL DE INVESTIGACIÓN Y DESARROLLO TECNOLÓGICO
SUBDIRECCIÓN ACADÉMICA

DR. CARLOS MANUEL ASTORGA ZARAGOZA
SUBDIRECTOR ACADÉMICO

C. c. p. Departamento de Ciencias Computacionales
Departamento de Servicios Escolares

CMAZ/CHG

Dedicatoria

Con cariño dedico este trabajo a:

Dios, por darme fuerzas cada día, guiar mis pasos
y permitirme cumplir una meta más en la vida.

A mis padres, Carlos Moreno y Erika Calderón,
por los esfuerzos que realizan cada día por mí,
por las enseñanzas que me han brindado,
por sus cansancios y desvelos,
por los regaños y el amor que me tienen;
Dios les pague todo lo que han hecho por mí.
¡Los amo!

A mis hermanos, Juan, Yoselin, Helem, Jaaziel y José,
por las risas, las travesuras, las aventuras y peleas,
he aprendido mucho con ustedes, Dios les pague.

Agradecimientos

Mi más profundo agradecimiento a mi director de tesis, el Dr. Joaquín Pérez Ortega, por brindarme el apoyo y la guía en esta etapa de mi formación profesional, y tener paciencia y confianza en mí para la realización de este trabajo de investigación.

A los miembros de mi comité tutorial: Dra. Alicia Martínez Rebollar y Dr. Javier Ortiz Hernández, por sus consejos y observaciones brindadas en el desarrollo de este trabajo de investigación.

A la Dra. Leticia Sánchez Lima, muchas gracias por sus enseñanzas y consejos, su apoyo en la redacción es muy valiosa.

Al Consejo Nacional de Ciencia y Tecnología (CONACYT) y al Centro Nacional de Investigación y Desarrollo Tecnológico (CENIDET), por la oportunidad y facilidad brindada para la realización de esta tesis.

Al gran equipo de trabajo conformado por: Andrea, Sandra, Nancy, Gerardo, César, muchas gracias por el apoyo brindado y la convivencia.

A mis amigos: Jorge, Dianely, Erick, Regino y Alejandra, por las risas y desvelos, han hecho que el tiempo que pasé aquí sea una gran experiencia.

A mi tía Alejandra Moreno, muchas gracias por haberme apoyado y orientado para llegar hasta aquí, la quiero mucho.

A mis mejores amigos: Cesar Gilberto y Wilson Alexis, por el apoyo brindado, los consejos, regaños y sobre todo las carcajadas; ¡si yo pude, ustedes también!

A mi mejor amiga Liz, muchas gracias por las horas de platica, risas y tristezas, porque aunque estemos lejos, siempre preguntas por mí, ¡ya me siento realizado!

A Rocío, por su amor, apoyo y paciencia; me has motivado a no rendirme, a superar cada reto que da la vida y sobre todo a no olvidar el camino de Dios.

¡Muchas gracias!

Resumen

En esta investigación se aborda el problema de evaluación de algoritmos. En particular la evaluación de variantes del algoritmo *K-Means* para resolver grandes instancias.

Desde el surgimiento de la familia de algoritmos *K-Means*, se han realizado diversos estudios para mejorar algunas de sus etapas y con ello reducir el costo computacional. Se ha observado en la literatura especializada que las variantes propuestas obtienen mejores resultados en cuanto a tiempo o calidad de solución frente al algoritmo estándar o incluso frente a otras variantes. Sin embargo, el diseño de experimentos que aplican abarca pocas instancias, y con ello surge la incertidumbre sobre la eficiencia y eficacia que podrían tener estas variantes al resolver otro tipo de instancias.

En este trabajo de investigación, se propone la aplicación de dos variantes prometedoras en la solución de diferentes tipos de instancias, mayormente sobre instancias grandes. La variante *Fahim* cuya propuesta es ampliamente citada y reconocida por la exclusión de cálculos de distancia de centroides a objetos, lo cual genera un ahorro en el costo computacional. La variante *O-K-Means* es una propuesta relevante que genera menor costo computacional por realizar la convergencia del algoritmo cuando el total de objetos que cambian de grupo en una iteración es menor a un umbral definido.

Esta propuesta se validó realizando pruebas de ejecución con el algoritmo *K-Means* estándar. Como referencia se implementó el algoritmo propuesto por Lloyd en 1982. Además de implementar las variantes *Fahim* y *O-K-Means*. Se realizó un diseño de experimentos donde se usaron 37 conjuntos de datos reales de repositorios reconocidos, 23 instancias fueron clasificadas como pequeñas y 14 como instancias grandes. Se realizaron comparaciones de tiempo y calidad de solución con respecto al algoritmo *K-Means* estándar. Los resultados obtenidos muestran que la variante *Fahim* es dominante al resolver instancias grandes mejorando la calidad hasta un 0.71% en el mejor de los casos. La variante *O-K-Means* demostró ser dominante con las instancias grandes al resolverlas en menor tiempo hasta un 93.57% en el mejor de los casos.

Finalmente, se considera que esta investigación aporta beneficios a investigadores o usuarios que buscan resolver instancias similares a las que se usaron en esta investigación, ofreciéndoles una caracterización de la variante que se adecue mejor a sus necesidades.

Abstract

In this research it is addressed the problem of algorithm selection. In particular the selection of variants of the K-Means algorithm to solve large instances.

Since the emergence of the K-Means family of algorithms, several studies have been carried out to improve some of its stages and thus reduce the computational cost. It has been observed in the specialized literature that the proposed variants obtain better results in terms of time or solution quality compared to the standard algorithm or even compared to other variants. However, the design of experiments that they apply covers few instances, and with this arises the uncertainty about the efficiency and effectiveness that these variants could have when solving other types of instances.

In this research work, we propose the application of two variants that are promising in solving different types of instances, mostly on large instances. The Fahim variant whose proposal is mostly cited and recognized by the exclusion of centroid distance calculations to objects, which generates a saving in computational cost. The O-K-Means variant whose proposal is relevant and generates lower computational cost by converging the algorithm when the total number of objects that change group in an iteration is less than a defined threshold.

This proposal was validated by performing execution tests with the standard K-Means algorithm. As a reference, the algorithm proposed by Lloyd in 1982 was implemented. In addition to implementing the Fahim and O-K-Means variants. A design of experiments was performed where 37 real instances of recognized repositories were used, 23 instances were classified as small and 14 as large instances. Time and solution quality comparisons were performed with respect to the standard K-Means algorithm. The results obtained show that the Fahim variant is dominant in solving large instances improving the quality up to 0.71% in the best case. The O-K-Means variant proved to be dominant with large instances by solving them in less time up to 93.57% in the best case.

Finally, this research is considered to bring benefits to researchers or users seeking to solve instances similar to those used in this research, offering them a characterization of the variant that better suits their needs.

Contenido

Pág.

Lista de tablas x

Lista de figuras x

Capítulo 1: Introducción 1

1.1 Contexto de la investigación 2

1.2 Planteamiento del problema 3

1.3 Hipótesis 3

1.4 Objetivo general 4

1.4.1 Objetivos específicos 4

1.5 Justificación 4

1.6 Alcances 5

1.7 Limitaciones 5

Capítulo 2: Revisión del estado del arte 6

2.1 Mejoras en la fase de inicialización 7

2.1.1. *Efficient and Fast Initialization Algorithm for K-Means Clustering* 7

2.1.2. *Improvement of K-Means algorithm based on density* 8

2.2 Mejoras en la fase de clasificación 9

2.2.1. Una heurística eficiente aplicada al algoritmo *K-Means* para el agrupamiento de grandes instancias altamente agrupadas 9

2.2.2. *A-means: improving the cluster assignment phase of K-Means for Big Data* 10

2.2.3. *An efficient enhanced K-Means clustering algorithm* 10

2.2.4. *Optimization of the K-Means algorithm for the solution of high dimensional instances* 11

2.2.5. Mejora del algoritmo K-Means mediante una meta-heurística orientada a la reducción de su complejidad computacional 12

2.2.6. *A Time-Efficient pattern reduction algorithm for K-Means clustering* 13

2.3 Mejoras en la fase de convergencia 14

2.3.1. *The early stop heuristic: A new convergence criterion for K-Means* 14

2.3.2. *Balancing effort and benefit of K-Means clustering algorithms in relamns Big Data realms* 14

2.3.3. *Comparison of a time efficient modified K-mean algorithm with K-Mean and K-medoid algorithm* 15

2.4 Mejoras en distintas fases del algoritmo 15

2.4.1. Desarrollo de heurísticas para la mejora del algoritmo *K-Means* en las fases de clasificación y convergencia 15
2.4.2. Desarrollo de una mejora al algoritmo *K-Means* orientada al paradigma de *Big Data* 17
2.4.3. Optimización del algoritmo *K-Means* orientada a *Big Data* mediante la integración de heurísticas en las fases de clasificación y convergencia 18
2.4.4. *Optimized big data K-Means clustering using MapReduce* 19
Capítulo 3: Estudio y clasificación de instancias 22
3.1 Instancias reconocidas 23
3.1.1 Conjunto de datos *Iris* 23
3.1.2 Conjunto de datos *Appendicitis* 23
3.1.3 Conjunto de datos *Newthyroid* 24
3.1.4 Conjunto de datos *Glass identification* 24
3.1.5 Conjunto de datos *Bupa* 24
3.1.4 Conjunto de datos *Wine* 24
3.1.7 Conjunto de datos *Ecoli* 24
3.1.8 Conjunto de datos *Balance scale* 24
3.1.9 Conjunto de datos *Heart disease* 25
3.1.10 Conjunto de datos *Cleveland* 25
3.1.11 Conjunto de datos *Steel plates faults* 25
3.1.12 Conjunto de datos *Pima* 25
3.1.13 Conjunto de datos *Pima indians* 25
3.1.14 Conjunto de datos *Breast cancer wisconsin* 25
3.1.15 Conjunto de datos *Concrete data* 26
3.1.16 Conjunto de datos *Cloud* 26
3.1.17 Conjunto de datos *Spectf* 26
3.1.18 Conjunto de datos *Yeast* 26
3.1.19 Conjunto de datos *Vehicle silhouettes* 26
3.1.20 Conjunto de datos *Abalone* 26
3.1.21 Conjunto de datos *Image segmentation* 27
3.1.22 Conjunto de datos *Page blocks clasification* 27
3.1.23 Conjunto de datos *Wine quality* 27
3.1.24 Conjunto de datos Parkinsons 27

3.1.26 Conjunto de datos *Pen digits* 27
3.1.27 Conjunto de datos *Magic gamma telescope* 28
3.1.28 Conjunto de datos *Isolet* 28
3.1.29 Conjunto de datos *Landsat satellite* 28
3.1.30 Conjunto de datos *Letter recognition* 28
3.1.31 Conjunto de datos *Optical digits.* 28
3.1.32 Conjunto de datos *Shuttle* 28
3.1.33 Conjunto de datos *Person activitis* 29
3.1.34 Conjunto de datos *Musk clean* 29
3.1.35 Conjunto de datos *Corel image features* 29
3.1.36 Conjunto de datos *Bag of words* 29
3.1.37 Conjunto de datos *Covertype* 29
3.2 Clasificación 30
Capítulo 4: Experimentación y análisis de resultados 32
4.1 Selección de variantes de *K-Means* 32
4.2 Diseño experimental 33
4.3 Dominio de las variantes en el tiempo 34
4.4 Dominio de variantes en calidad 35
4.5 Desempeño de variantes con instancias pequeñas 37
4.6 Desempeño de variantes con instancias grandes 40
4.7 Comparación de dominio entre variantes 42
Capítulo 5: Conclusiones y trabajos futuros 45
5.1 Conclusiones 45
5.2 Trabajo futuro 47
Referencias 48
Anexo A: Resultados experimentales con variante *Fahim* 52
Anexo B: Resultados experimentales con variante O-K-Means 59

Lista de tablas

Tabla 1. Características principales en los trabajos relacionados. 21
Tabla 2. Concentrado de instancias reales pequeñas. 30
Tabla 3. Concentrado de instancias reales grandes. 31
Tabla 4. Desempeño de las variantes Fahim y O-K-Means con instancias pequeñas. 39
Tabla 5. Desempeño de las variantes Fahim y O-K-Means con instancias grandes. 41
Tabla 6. Diferencia de dominio en el tiempo entre la variante Fahim y O-K-Means. 42
Tabla 7. Diferencia de dominio en la calidad entre la variante Fahim y O-K-Means. 43

Lista de figuras

Figura 1. Solución del problema. 4
Figura 2. Distribución de instancias sintética y reales: a) Instancia sintética bidimensional con n = 10,000 distribuido en rango de 1–100; b) Instancia real iris con n = 150 y d = 2. 23
Figura 3. Porcentaje de dominio de las variantes Fahim y O-K-Means con la métrica de tiempo. Experimentos A y B. 35
Figura 4. Porcentaje de dominio de las variantes Fahim y O-K-Means con la métrica de calidad. Experimentos A y B. 37

Capítulo 1

Introducción

El agrupamiento de datos se ha convertido en una necesidad hoy en día, esto es respecto al crecimiento constante de información que se da día a día [1]. Una forma de realizar el agrupamiento es mediante el uso de algoritmos y, uno de ellos es *K-Means*. Este algoritmo es muy usado por su facilidad de implementación computacional, aunque su procesamiento tiene una limitación ya que tiene una alta complejidad computacional [2].

La problemática del agrupamiento ha generado la realización de estudios enfocados a la mejora del algoritmo *K-Means* en sus distintas etapas: inicialización, clasificación, cálculo de centroides y convergencia, algunos son los artículos por Pérez en [3] y [4] estas mejoras están enfocadas a ciertos problemas computacionales, los cuales son dirigidos a la eficiencia del algoritmo o la capacidad de solución de instancias de datos [5].

En la investigación desarrollada por Basave [6] se realizó la primera mejora al algoritmo *K-Means* en el Centro Nacional de Investigación y Desarrollo Tecnológico (CENIDET). También se ha realizado la aplicación del algoritmo para la solución de ciertos problemas de agrupamiento [7-9]. Estas investigaciones tienen como relevancia generar resultados

comparativos con el algoritmo estándar de *K-Means* o de otra heurística del estado del arte, representando con ello resultados prometedores.

La presente investigación tiene como objetivo generar una comparación de las características que describen las mejoras realizadas al algoritmo *K-Means* del estado del arte, que sean relevantes y a su vez aplicarlas a instancias grandes. El propósito es documentar qué mejora se debe utilizar al solucionar un tipo de instancia, describiendo con ello que es la mejora dominante sobre otras.

1.1 Contexto de la investigación

En el CENIDET se ha realizado investigación para mejorar y comparar el algoritmo *K-Means* frente a muchas de sus variantes, como resultado se han reportado diversas tesis y publicado diversos artículos en revistas y conferencias. Algunos de ellos, se describen a continuación:

a) En la tesis de maestría [6] "Mejoramiento de la eficiencia y eficacia del algoritmo de agrupamiento *K-Means* mediante una nueva condición de convergencia" se describe una mejora al algoritmo *K-Means* en la fase de convergencia. Esta mejora se enfoca en detener el algoritmo cuando se identifique un óptimo local, teniendo como base la comparación del error al cuadrado en dos iteraciones sucesivas y comparar la posición de los centroides en dos iteraciones sucesivas.
b) En la tesis de maestría [7] "Estudio e implementación de las mejoras más relevantes del algoritmo *K-Means* y su análisis comparativo" describe la integración de varias mejoras al algoritmo *K-Means* en un sistema denominado *H-Kmeans*. Realiza la agrupación seleccionando el conjunto de datos a resolver, la cantidad de grupos y las mejoras a usar en cada etapa del algoritmo.
c) En la tesis doctoral [8] "Desarrollo de heurísticas para la mejora del algoritmo *K-Means* en las fases de clasificación y convergencia" se hace uso del algoritmo *K-Means* para incrementar su eficiencia. Es así como explica la creación de dos heurísticas que mejoran las etapas de clasificación y convergencia del algoritmo, en la cual la primera determina si un objeto esta fuera del umbral de equidistancia, esta conserva su grupo; el segundo describe un umbral de paro, en el cual el total de objetos que cambian de grupo es menor al umbral, conservan su grupo y la convergencia se realiza en menos iteraciones.

d) En la tesis de maestría [9] “Desarrollo de una mejora al algoritmo *K-Means* orientadas al paradigma de Big Data” se describe la implementación de una metodología para la comparación de mejoras al algoritmo *K-Means* en su fase de clasificación. Asimismo, describe la implementación de tres variantes del algoritmo y validando el dominio que presentan las mejoras sobre la eficacia y eficiencia.

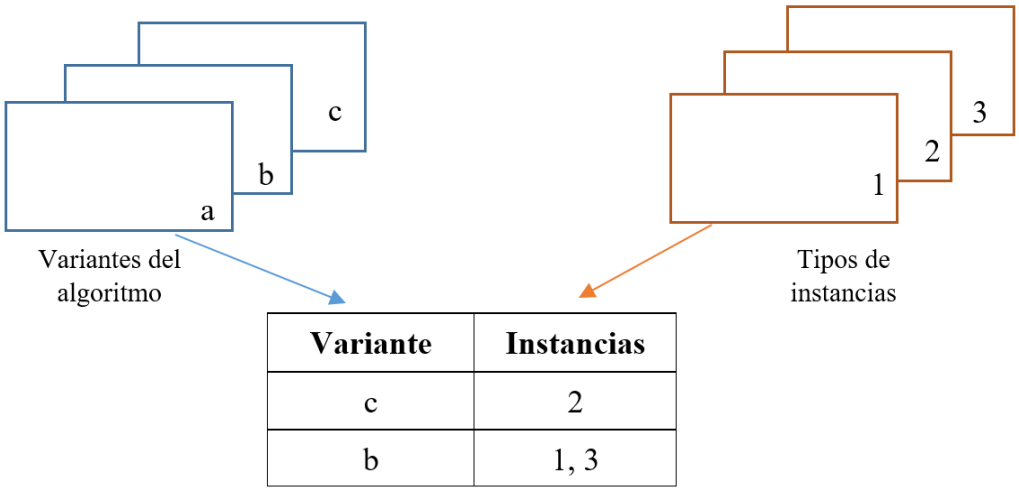
1.2 Planteamiento del problema

K-Means es uno de los algoritmos de agrupamiento más utilizados, ya que es relativamente sencillo interpretar sus resultados [5]. Sin embargo, tiene un alto costo computacional [2]. Para reducir la complejidad computacional se han propuesto múltiples mejoras al algoritmo en sus diferentes etapas. Algunas de las variantes han mostrado ser mejores que otras para algún tipo de instancia particular. Debido a que existen numerosas variantes del algoritmo y diferentes tipos de instancias a resolver, surge el problema de que, para una instancia dada, cómo seleccionar la variante del algoritmo que la pueda resolver mejor. En este trabajo de investigación se propone caracterizar las principales mejoras a *K-Means* e identificar cual variante es dominante sobre otras al resolver un tipo de instancia.

1.3 Hipótesis

Dado que existen diversas variantes del algoritmo *K-Means*, es posible determinar cuál de esas variantes es dominante al resolver un tipo de instancia, mediante comparación de resultados.

Como se muestra en la Figura 1, se tiene un conjunto de variantes y diferentes instancias, entre ellas se realizarán pruebas para determinar que variante resuelve mejor un tipo de instancia.



Variante	Instancias
c	2
b	1, 3

Figura 1. Solución del problema.

1.4 Objetivo general

Caracterizar dos mejoras del algoritmo *K-Means* estándar que mejoren la solución de grandes instancias de datos.

1.4.1 Objetivos específicos

1) Seleccionar las mejoras más relevantes a *K-Means* en sus distintas fases.
2) Realizar una selección de las distintas instancias con mayor número de datos, como los usados por *Big Data*.
3) Caracterizar las mejoras al algoritmo *K-Means* en la solución de las instancias.
4) Seleccionar las mejoras más prometedoras a partir de la incorporación de calidad de solución y reducción de tiempo.
5) Desarrollar y documentar una taxonomía de tipos de instancias y los algoritmos dominantes sobre ese tipo de instancias.

1.5 Justificación

Actualmente existen muchos algoritmos de agrupamiento. Algunos de ellos se describen en [10] dedicados a la solución de ciertos tipos de instancias como minería de datos o *big data*, entre estos algoritmos está *K-Means* al cual se le han realizado varias mejoras para su implementación en distintos problemas de agrupación, habiendo varias mejoras de *K-Means* como son los artículos [11-13] y ser uno de los algoritmos más usados, puede que al utilizar una variante no se obtengan los resultados esperados, teniendo en cuenta que otras variantes pueden generar mejores resultados.

En esta propuesta, realizar la caracterización de las mejoras a *K-Means* puede ser de gran ayuda a cualquier investigador o persona que recurra al uso del algoritmo para realizar agrupación de datos, generando una selección adecuada respecto al tipo de instancia que se requiera solucionar, y evitar generar un mayor costo computacional, pérdida de calidad o incluso mayor tiempo de solución.

1.6 Alcances

- Obtener un análisis comparativo de las dos mejoras hechas al algoritmo *K-Means* estándar.
- Implementación de dos mejoras a *K-Means*.
- Análisis de resultados con instancias reales pequeñas y grandes
- Identificación del dominio de las variantes sobre las instancias.

1.7 Limitaciones

- Se implementaron dos mejoras del algoritmo *K-Means* estándar de los últimos 15 años: artículos e investigaciones de tesis.
- Se probaron las mejoras con instancias reales de repositorios reconocidos.
- Las pruebas se realizaron con equipo de hardware y software disponible en el CENIDET y equipo personal.

Capítulo 2

Revisión del estado del arte

El algoritmo *K-Means* es muy conocido, además existen numerosas publicaciones que describen su funcionamiento y otras diferentes aportando una mejora a algunas de sus etapas. Con el fin de generar mayor eficiencia y eficacia en la agrupación de datos. Estas mejoras han sido probadas y comparadas frente a otras variantes para demostrar que una de ellas puede ser dominante en un determinado concepto como el esfuerzo computacional, el tiempo de procesamiento, entre otros.

En este apartado se dará a conocer algunas publicaciones relevantes, que se tomaron como apoyo para la realización de esta tesis. El propósito es, conocer las variantes o mejoras del algoritmo *K-Means*, conocer qué estrategias se aplican para realizar pruebas frente a grandes instancias y con ello generar una buena base experimental.

En sección 2.1 se describen las publicaciones que abordan mejoras al algoritmo *K-Means* en su etapa de inicialización. En la sección 2.2 se describen las publicaciones que abordan mejoras al algoritmo *K-Means* en su etapa de clasificación. En la sección 2.3 se describen las publicaciones que abordan mejoras al algoritmo *K-Means* en su etapa de

convergencia. En la sección 2.4 se describen las publicaciones que abordan mejoras en más de una etapa del algoritmo *K-Means*.

2.1 Mejoras en la fase de inicialización

2.1.1. *Efficient and Fast Initialization Algorithm for K-Means Clustering*

En [14] se describe la propuesta de un nuevo algoritmo para la inicialización de *K-Means*, esto es con respecto a que el algoritmo *K-Means* es sensible para la inicialización de los centroides, además puede converger a un mínimo local. Para la generación de esta propuesta se utilizó un marco de agrupación visual para el uso de *K-Means* estándar y el algoritmo propuesto llamado *ElAgha initialization*.

En el marco de visualización se usaron tres métodos de inicialización, los cuales son inicialización aleatoria, inicialización manual e *ElAgha initialization,* ésta última genera *K* puntos usando una técnica semi-aleatoria. Hace que la diagonal de los datos sea una línea de partida y selecciona los puntos aleatoriamente a su alrededor. *ElAgha initialization* primero encuentra límites entre los datos, luego genera divisiones en el área donde se encuentran los puntos de datos formando *K* filas y *K* columnas. *ElAgha initialization* usa la esquina superior izquierda de las celdas que se encuentran en la diagonal como puntos de base para inicializar los centroides y comenzar su distribución a través de las iteraciones, pero solo se pueden distribuir dentro de sus respectivas celdas.

Se realizaron pruebas de comparación entre el método estándar de inicialización y *ElAgha initialization,* el propósito es verificar que este método es eficaz y eficiente a comparación del método estándar. Los métodos fueron puestos a prueba con diez ejecuciones para la ver la comparación, se usaron tres conjuntos de datos artificiales y un conjunto de datos reales llamado iris. El primer conjunto de datos consiste en 320 puntos y ocho grupos, el segundo conjunto de datos consiste en 373 puntos y tres grupos, y el tercer conjunto de datos consiste en 211 puntos y seis grupos.

Después de que se realizaran las pruebas, el primer conjunto de datos obtuvo un índice de error promedio del 65%, pero *ElAgha initialization* obtuvo solo 2.27%; el segundo conjunto de datos obtuvo un índice de error promedio de 43.4%, mientras que *ElAgha initialization* obtuvo solo el 5%; el tercer conjunto de datos obtuvo un índice de error promedio de 56.47% y *ElAgha initialization* obtuvo 4.44%.

Después de realizar las pruebas y comparar resultados, se llegó a la conclusión en que *ElAgha initialization* tiene un margen de error menor a la inicialización tradicional, ayudando a que la agrupación de los datos sea más eficaz y eficiente.

2.1.2. *Improvement of K-Means algorithm based on density*

En [15] se presenta una mejora al algoritmo *K-Means* llamado *D-K-Means* que adopta el concepto de número de densidad. Se considera que el número del tamaño de la densidad se debe extraer del conjunto de datos original, y el punto en el centro de cada grupo es considerado punto central del grupo. Para realizar la agrupación implementan un método llamado puntos centrales geométricos, esto con la finalidad de actualizar en cada iteración los puntos centrales del grupo en puntos de alta densidad.

Esta publicación afirma que el método de inicialización de *K-Means* es deficiente por seleccionar puntos centrales aleatorios. La propuesta de este artículo se basa en calcular la densidad de datos iniciales y extraer el conjunto de puntos con números de alta densidad que cumplan las condiciones. Los métodos para seleccionar puntos centrales se concentran en los puntos de alta densidad, el punto central del grupo se selecciona en el rango de puntos de alta densidad

El método realiza siete pasos, el primero se describe como análisis inicial de datos y selección en el cual se clasifican *N* puntos de datos por la densidad de sus puntos, extraer puntos de alta densidad y rechazar los puntos de baja densidad. El segundo paso es la selección inicial del punto central en el cual el conjunto de datos de alta densidad se le hace una selección aleatoria y de ese primer punto se hace una segunda sección al dato más alejado de ese punto y ese se considera el segundo punto central del grupo. El tercer paso realiza el cálculo de distancia de cada dato al centro del grupo inicial. El cuarto paso realiza el cálculo de dos puntos con la mayor distancia y se usa el punto medio de estos dos para ser el nuevo punto central. El quinto paso es repetir los dos pasos anteriores hasta que la convergencia sea alcanzada. Los últimos dos pasos están enfocados en calcular un punto central final del grupo y describe que su agrupamiento es más preciso que el método que aplica el algoritmo *K-Means* estándar.

Para las pruebas de esta propuesta se utilizó lenguaje *MATLAB* para la implementación del algoritmo estándar *K-Means* y *D-K-Means*. El conjunto de datos de prueba usado se

denomina *Iris* el cual contiene 150 objetos con 4 atributos. Para evaluar los resultados se compara la calidad usando el índice de precisión de clasificación (*DB*) y el espaciado entre clases (*SP*), cuanto menor sea el resultado de *DB* mayor será la relación de los datos del mismo grupo, y mientras más alto sea el valor de *SP* mayor será la diferencia entre las clases.

Los valores de los resultados fueron favorables, *D-K-Means* obtuvo una curva de valor más estable que *K-Means* estándar en el índice *DB* y *SP*. Se demostró que, con los resultados obtenidos, la precisión de la propuesta genera resultados de calidad favorables y mejores a comparación del algoritmo *K-Means* estándar.

2.2 Mejoras en la fase de clasificación

2.2.1. Una heurística eficiente aplicada al algoritmo *K-Means* para el agrupamiento de grandes instancias altamente agrupadas

Esta mejora descrita en [1] está enfocada a la fase de clasificación del algoritmo *K-Means* haciendo que el cálculo de distancia se relacione solo con los grupos vecinos del grupo al cual ya pertenece un objeto, descartando de esta manera grupos más lejanos.

La heurística propuesta se llamó Panal de abeja y describe una estructura de grupos creados por el algoritmo *K-Means*, generando en la primera iteración grupos con formas de polígonos irregulares, pero al llegar a la última iteración se visualizaron grupos con formas de polígonos regulares.

Las pruebas fueron realizadas en cinco procesos y cada una con 30 ejecuciones comparando resultados de la heurística con el algoritmo estándar *K-Means*, en el primero se usó una instancia de 25,000 objetos distribuidos de forma aleatoria en dos dimensiones y para el resto se usaron variaciones objetos, grupos y dimensiones.

Los resultados obtenidos sobre la heurística propuesta fue una reducción de tiempo de 90%, cabe mencionar que su eficiencia es mayor con el uso de instancias de más de 3,000 objetos, con más de 10 dimensiones y con el uso de 100 o más grupos, la calidad de solución del algoritmo fue reducción del 1%.

Tras comparaciones con las propuestas hechas por [14] y [16] respecto a *K-Means* estándar, se concluyó que la heurística propuesta tiene mayor impacto en instancias altamente agrupadas y con ello resulta útil para instancias grandes como es en el campo de la *big data*.

2.2.2. *A-means: improving the cluster assignment phase of K-Means for Big Data*

En [17] propone una mejora de la heurística propuesta en [13] para la solución de instancias de grandes objetos, la propuesta está enfocada en reducir los cálculos de distancia en la etapa de clasificación, esto se logra a través del cálculo de distancia de un objeto a sus dos centroides más cercanos, se obtiene una probabilidad de cambio del objeto a otro grupo y si este es menor al definido por el índice de equidistancia, se excluye su cálculo en iteraciones futuras.

Esta propuesta fue nombrada *A-Means* y en ella se usaron dos instancias reales en las cuales se describieron 1,140,000 y 1,000,000 de objetos respectivamente, 45 y 11 dimensiones respectivamente, siete instancias sintéticas en las cuales se describieron cinco de ellas contenían 2,000,000 objetos y el resto 1,000,000 objetos además que contenían de dos a siete dimensiones.

Se realizaron pruebas en las cuales se hicieron comparaciones entre el algoritmo *K-Means* estándar, *Early classification* [18] y *A-Means*, con base en los resultados se obtuvo que *A-Means* tiene mejores resultados en la reducción de tiempo de 98% y una reducción en la calidad de la solución de un 3%. Adicionalmente, se observó que *A-Means* tiende a ser más eficaz con el uso de mayor número de objetos y dimensiones, concluyendo que su uso puede ser prometedor con instancias en el campo de *big data*.

2.2.3. *An efficient enhanced K-Means clustering algorithm*

En [19], se presenta una propuesta para la fase de clasificación del algoritmo *K-Means*. La propuesta está enfocada en guardar información de cada iteración del algoritmo para aplicarlo a la siguiente iteración. La propuesta empleada se le denominó algoritmo *K-Means* mejorado.

El algoritmo *K-Means* realiza la agrupación de objetos y los muestra gráficamente junto con sus respectivos centroides. A través de cada iteración los centroides van cambiando su ubicación para estar lo más cerca posible de los objetos a los cuales pertenecerán. La propuesta describe que es necesario realizar un cálculo de distancia de cada objeto hacia su respectivo centroide en todas las iteraciones que realice el algoritmo. Al obtener la distancia de una iteración se compara con la nueva distancia de la siguiente iteración, si la nueva distancia es menor o igual que la distancia anterior, el punto permanece en su grupo y no hay necesidad de calcular sus distancias a los otros centros del grupo.

La propuesta empleada tiene la finalidad de ahorra tiempo requerido para calcular las distancias de los objetos hacia los centroides. Se ha visualizado que en cada iteración los centroides cambian de posición a medida que se integran o desintegran del grupo algunos objetos. Los objetos que están más cercanos del centroide no se les realiza el cálculo de su distancia ya que tienen bajas posibilidades de cambiar de grupo. Sin embargo, los que están más alejados de los centroides es necesario realizarles el cálculo de distancia a otros centroides y se asignará al grupo al que se encuentre más cerca.

Para conocer la eficiencia de la mejora se realizaron pruebas comparativas con el algoritmo *K-Means* estándar, *Overlapped K-Means* y *CLARA* [20], usando tres conjuntos de datos reales y uno sintético. El primer conjunto de datos llamado: *Letter image* contiene 20,000 objetos con 16 atributos, y obtuvo un tiempo de 60 segundos al ser agrupado con 100 grupos, a diferencia de los otros algoritmos que obtuvieron un tiempo más tardado. El conjunto de datos llamado: *Abalone* contiene 4,177 objetos con ocho atributos y la mejora propuesta realizó su agrupación con un total de 1,000 grupos requeridos en un tiempo de 10 segundos, a diferencia de los otros algoritmos que realizaron la agrupación en tiempo más tardado. El conjunto de datos llamado: *Wind* contiene 6,574 objetos con 15 atributos y la mejora propuesta realizó la agrupación con 160 grupos requeridos en un tiempo de 17 segundos, demostrando ser más eficiente que los otros algoritmos.

Con las pruebas realizadas se llegó a la conclusión de que la propuesta implementada mejora los tiempos de agrupación del algoritmo *K-Means*, eso sin perder la calidad de agrupación que en las pruebas fueron relativamente similares a los otros algoritmos. Con ello la propuesta es mejor en eficiencia respectivamente a los algoritmos *K-Means* estándar, *CLARA* y *Overlapped K-Means.*

2.2.4. *Optimization of the K-Means algorithm for the solution of high dimensional instances*

En [5] está enfocada en reducir la complejidad del algoritmo *K-Means* y su propuesta trabaja en el paso de clasificación del algoritmo.

Mediante la observación del comportamiento del algoritmo, los autores del artículo decidieron realizar una reducción a los cálculos de las distancias que hay entre los objetos y

los centroides, ya que en las observaciones se descubrió que los objetos pueden migrar a grupos adyacentes sin cruzar a grupos distantes.

Su experimentación se basó en el uso de instancias tridimensionales de distribución uniforme y de alta densidad, sus grupos formaron dodecaedros generando así cálculos de distancia a doce grupos adyacentes incluyendo el grupo al que pertenece el objeto.

En particular el enfoque de solución de esta investigación, logra obtener la reducción compleja del algoritmo mediante la mejora de agrupación de instancias dimensionales, pero para ello se propuso la determinación de un umbral para instancias con un número par de atributos como son entre dos y 200.

Su experimentación se puso a prueba con el algoritmo estándar de *K-Means* usándose 25,000 objetos para cada instancia y se encontró dos patrones de interés, en el primero se observó que el umbral variaba entre dos y 30 atributos a comparación del segundo que si se mantenía estable en atributos de 30 a 200.

Su rendimiento fue comparado y se caracterizó que en tiempo y calidad describiendo que la solución de calidad no pasaba el uno por ciento, en el tiempo de solución se encontró una reducción considerable de 96.51% y estos valores son comparados al tiempo y calidad de solución de *K-Means* estándar.

2.2.5. Mejora del algoritmo K-Means mediante una meta-heurística orientada a la reducción de su complejidad computacional

En [2] está orientada al aprovechamiento del rendimiento computacional por medio de una mejora al algoritmo *K-Means*, el cual detecta algunos grupos que se estabilizan y estos mismos son descartados en futuras iteraciones, generando así un mayor rendimiento.

La composición de esta investigación está dividida en dos heurísticas denominadas *H1* y *H2*.

La heurística *H1* se enfoca en crear dos subconjuntos, el primero es para aquellos objetos que tienen baja posibilidad de cambio de grupo y el segundo es para aquellos objetos que tiene alta posibilidad de cambio de grupo.

Le heurística *H2* se enfoca en observar los grupos que se vuelven estables en iteraciones tempranas esto quiere decir que sus objetos quedan estables en sus respectivos centroides y

por lo tanto una vez que se vuelven estables ya no se toman en cuenta en las siguientes iteraciones. Sin embargo, quedan vigentes hasta que todos los grupos se vuelven estables.

Esta mejora fue nombrada *N-Means* y se observó que obtuvo una reducción en tiempo de 91% y la disminución de calidad en un 5.5%, cabe mencionar que también se realizó una comparación con el algoritmo estándar *K-Means* usando instancias de prueba: *Birch1* y el *DIM*.

2.2.6. *A Time-Efficient pattern reduction algorithm for K-Means clustering*

En [16] se presenta una mejora al algoritmo *K-Means* realizando un trabajo llamado reducción de patrones, con el fin de reducir el tiempo de cálculo de los algoritmos de agrupación basados en *K-Means*.

El algoritmo propuesto funciona comprimiendo y eliminando en cada iteración patrones que es poco probable que cambien su membresía a partir de la iteración estable, la propuesta no solo es simple y fácil de implementar, sino que también se puede aplicar a muchos otros algoritmos de agrupación iterativos, como los algoritmos de agrupación basados en el núcleo y en la población.

Se realizaron experimentos con conjuntos de datos, de dos a 1,000 atributos y de 150 a 10,000,000 objetos, mismos que indican que con una pequeña pérdida de calidad, el algoritmo propuesto puede reducir significativamente el tiempo de cálculo de todos los algoritmos de agrupamiento de última generación, especialmente para grandes conjuntos de datos y de alta dimensión.

Los resultados obtenidos fueron realizados con base en 30 experimentos y diferentes números de grupos. Al compararlo con el algoritmo *K-Means* estándar, se observó una reducción de tiempo de 84% y en calidad de la solución una reducción de 64%.

2.3 Mejoras en la fase de convergencia

2.3.1. *The early stop heuristic: A new convergence criterion for K-Means*

Esta mejora descrita en [21] está centrada en el paso de convergencia del algoritmo *K-Means*, el cual se basa en observar el mayor desplazamiento de un centroide, de ello se obtiene un porcentaje de desplazamiento y se define un umbral de convergencia.

Para las pruebas se usaron ocho conjuntos de instancias, las primeras tres de datos sintéticos los cuales contienen 2,500, 10,000 y 40,000 objetos bidimensionales distribuidos uniformemente. Se usaron instancias reales con 1,030 objetos tridimensionales, 245,057 tridimensionales, 284,284 objetos tridimensionales, y las últimas dos con 414,528 y 657,308 objetos bidimensionales.

Los resultados obtenidos, con instancias sintéticas, fueron comparados con el algoritmo *K-Means* estándar. Se observó una reducción de tiempo del 83.68% y una reducción de calidad de la solución de 1.23%. Para las instancias reales, se obtuvo una reducción de tiempo de 87.06% y una reducción de calidad de la solución de 2.46%.

2.3.2. *Balancing effort and benefit of K-Means clustering algorithms in relamns Big Data realms*

En [4] se presenta una mejora del algoritmo *K-Means* denominada O-K-Means. Dicha mejora acelera el proceso de convergencia, parando al algoritmo cuando el total de los objetos que cambian de grupo en una iteración es menor a un umbral. Este valor, expresa una relación entre el esfuerzo computacional y la calidad de la solución.

Se realizaron pruebas computacionales aplicando el principio de Pareto. Las pruebas se realizaron con diferentes conjuntos de datos reales y sintéticos, en ellas fue necesario conocer el valor del umbral para determinar el número de iteraciones que debe realizar el algoritmo para converger. Los resultados fueron prometedores al observar una disminución en el tiempo de procesamiento, alcanzando 96.12% de reducción usando instancias reales y grandes, y 86.47% de reducción usando instancias reales pequeñas. A comparación del uso de instancias sintéticas, se observó que el algoritmo *O-K-Means* presentó mejor rendimiento que el algoritmo *K-Means* estándar, obteniendo un promedio de 96.10% de reducción tiempo de procesamiento y un promedio en la calidad de la solución del 0.5%

2.3.3. *Comparison of a time efficient modified K-mean algorithm with K-Mean and K-medoid algorithm*

En [12] se realizó una comparación de tiempo y calidad de solución usando una modificación del algoritmo *K-Means* estándar. El propósito fue generar un nuevo algoritmo que combinara los principios del algoritmo *K-Means* y *K-Medoid*.

La implementación parte de un algoritmo modificado llamado *Efficient K-Means*. Este algoritmo permite llegar a una solución óptima en cierta iteración y reducir la probabilidad de dividir un grupo de datos en dos o más debido a la adopción del criterio de parada. Para optimizar el tiempo y la calidad de solución se eligieron los centroides adecuados con los cuáles se hizo la agrupación, y se observó que los algoritmos de agrupación son sensibles a la inicialización. Se usaron conjuntos de datos reales y sintéticos para la prueba del algoritmo y comparación de los resultados.

La experimentación se realizó con valores de *K* entre dos y cinco. Se observó que a menor valor de *K* el tiempo de ejecución del algoritmo modificado es eficiente. Se observó que el algoritmo *K-Medoid* genera menos tiempo de ejecución a comparación del algoritmo *K-Means* estándar. También se observó que a igual número de objetos el algoritmo modificado toma un tiempo de ejecución similar al algoritmo *K-Medoid.*

Con los resultados obtenidos del método implementado, se concluyó que, al seleccionar un número de grupos considerables, el tiempo de ejecución y el rendimiento será menor al esperado. También la comparación muestra en qué momentos es considerable utilizar un cierto valor de *K* para alcanzar eficiencia en la ejecución del algoritmo y también menos esfuerzo computacional.

2.4 Mejoras en distintas fases del algoritmo

2.4.1. Desarrollo de heurísticas para la mejora del algoritmo *K-Means* en las fases de clasificación y convergencia

En la tesis [8] se desarrollaron dos heurísticas que abordan el problema de incrementar la eficiencia de *K-Means*. La primera heurística empleada se enfoca en la fase de clasificación del algoritmo K-*Means* y se denominó *A-Means*. La segunda heurística empleada se enfoca en la fase de convergencia del algoritmo *K-Means* y se denominó *O-K-Means*.

Dentro de la propuesta hay dos enfoques para la solución, el primero es realizar una modificación a uno de los conceptos del algoritmo *Early classification*. *A-Means* define el índice de equidistancia como: el valor absoluto de diferencia de distancias de un objeto a dos centroides que se encuentren cerca del mismo. *A-Means* define el umbral de equidistancia como: la suma de los dos desplazamientos mayores de los centroides en la iteración. *A-Means* usa estos dos conceptos para realizar una mejora a *Early Classification*, en el cual se describe si un objeto tiene una distancia más cercana al centroide que ya pertenece respecto a otro centroide, conserva su membresía.

Para el segundo enfoque de solución de la propuesta es la aplicación de la heurística denominada *O-K-Means*. Esta heurística está enfocada en la fase de convergencia del algoritmo *K-Means*, y su propósito es detener el algoritmo cuando el número de objetos que cambian de grupo es menor a un umbral definido.

Las pruebas realizadas con las heurísticas propuestas se hicieron en cuatro experimentos. El primer experimento es demostrar la calidad y el tiempo que resuelve *A-Means*, *Early Classification* y *K-Means* respectivamente entre ellos. Se utilizaron dos conjuntos de datos reales y ocho de datos sintéticos, en los cuales se obtuvo una reducción de tiempo de 70.44% y una pérdida de calidad de -1.44%.

En el segundo experimento se describe el uso de 14 instancias sintéticas de prueba para lograr ahorros de tiempo de ejecución. En el experimento se obtuvieron resultados prometedores, en el mejor de los casos se logró reducción de tiempo de hasta 97.59% y una pérdida de calidad de -0.24% respecto a *K-Means*.

En el tercer experimento se describe el resultado promedio de 30 ejecuciones sobre seis instancias sintéticas de prueba. En el experimento se obtuvieron resultados prometedores de los cuales, en el mejor de los casos, la reducción del tiempo fue de hasta 89.13% con una pérdida de calidad de -0.14%, ambos resultados son respectivos a los de *K-Means*.

En el cuarto experimento se realizaron ejecuciones con el algoritmo hibrido *A-Means/O-K-Means* respecto a *K-Means*. Se utilizaron dos instancias reales, de las cuales, se obtuvieron resultados prometedores. Se realizaron 30 ejecuciones de las cuales se obtuvo el promedio de resultados con 98% en reducción de tiempo y pérdida de calidad de hasta -2.90%.

Con los experimentos realizados se concluye en que las heurísticas propuestas son mejores en cuanto al incremento de la eficiencia del algoritmo *K-Means*. Es conveniente el uso de las heurísticas para aplicarlo a instancias dentro del paradigma de *big data* ya que, al ser puestos a prueba, la reducción de tiempo fue menor al que resuelve el algoritmo *K-Means*.

2.4.2. Desarrollo de una mejora al algoritmo *K-Means* orientada al paradigma de *Big Data*

En la tesis [9] se propone una mejora al algoritmo de agrupación *K-Means* llamada *H-Kmeans*. La propuesta está enfocada en procesar una instancia dada mediante la composición de mejoras en distintas fases del algoritmo, es por ello que se le denominó *H-Kmeans* por ser una mejora híbrida del algoritmo *K-Means*.

La propuesta incluye mejoras en tres de las cuatro fases del algoritmo que son de inicialización, clasificación y convergencia. Para las mejoras integradas en la fase de inicialización se seleccionó las mejoras denominadas "*ElAgha*" [14] y "*K-Means*++" [22]. Para las mejoras integradas en la fase de clasificación se seleccionó las mejoras denominadas "*Early Classification*" [18] y "*Enhanced K-Means*" [19]. Por último en la fase de convergencia las mejoras seleccionadas se denominan "*Early Stop*" y "*Early Stop Heuristic*".

Para la metodología propuesta se generaron 4 pasos a seguir en el proceso del agrupamiento. El primer paso es la selección de instancia y la cantidad de grupos en la que se desea realizar la agrupación, el segundo paso realiza la selección del algoritmo *K-Means* o la mejora a combinar por fase, el tercer paso se realiza la configuración de los parámetros seleccionados y realiza el procesamiento de la instancia, y por último, se generan dos documentos con extensión ".txt" en cual contienen la información resultante del procesamiento.

Para poner a prueba esta mejora se utilizaron cuatro conjuntos de datos reales. La primera instancia denominada *Wind* contiene 126,692 objetos con seis atributos para k= 50 y 100. En esta prueba se alcanzó una reducción de tiempo de hasta 90% y en la calidad de la solución de 7.0% comparado con el *K-Means*. La segunda instancia denominada "*3D road network*" contiene 434,874 objetos y cuatro atributos en el cual se alcanzó una reducción de tiempo de hasta 93.5% y un porcentaje de calidad de la solución de 39.9% mejor que *K-Means*. La tercera instancia denominada "*Household power consumption*" contiene 2,049,280 objetos y cuatro atributos en el cual se alcanzó una reducción de tiempo de hasta 97.3% y porcentaje

de calidad de solución 3.5% mejor que *K-Means*. La cuarta instancia denominada "*Letter recognitio*n" contiene 20,000 objetos con 16 atributos, alcanzó una reducción de tiempo de hasta 66.6% y calidad de solución de 0.4% mejor que *K-Means*. Se observó que con la heurística híbrida se alcanzaron reducciones de tiempo y calidad de solución prometedoras.

2.4.3. Optimización del algoritmo *K-Means* orientada a *Big Data* mediante la integración de heurísticas en las fases de clasificación y convergencia

En la tesis [23] se desarrolló dos mejoras al algoritmo *K-Means*. La primera está enfocada al paso de clasificación y la segunda al paso de convergencia. La finalidad de esta investigación es reducir el tiempo de ejecución del algoritmo. La implementación al paso de clasificación es reducir la cantidad de objetos con los cuales se realiza los cálculos, en cambio el paso de convergencia se encarga de terminar el algoritmo cuando la calidad de solución es cercana a la calidad final, aunque la pérdida de calidad sea menor.

La realización de las mejoras consiste en generar un algoritmo híbrido usando como base el comportamiento de los algoritmos *Enhanced K-Means* y *Early classification*, como resultado se integró la idea principal donde el primer algoritmo expresa: si un objeto alcanza una distancia menor al centroide a comparación de una iteración anterior, su cálculo de distancia se cancela; la idea principal del segundo algoritmo es que los objetos que se encuentren cercanos al borde del grupo tienen altas posibilidades de cambio de grupo y por lo tanto solo a ellos se les realizará el cálculo de distancia a los centroides.

La experimentación y validación de esta propuesta se dividió en dos secciones, la primera fue usando instancias sintéticas y la segunda usando instancias reales. Los resultados obtenidos fueron comparados con el algoritmo *K-Means* estándar, *Enhances K-Means* y *Early clasification*. Los resultados fueron prometedores en términos de tiempo, es conveniente mencionar que los resultados del algoritmo propuesto son similares al algoritmo Early clasification y en el mejor de los casos se obtuvo 94.75% en la reducción del tiempo. En cuanto a la reducción de calidad de solución, se determinó que los valores obtenidos del algoritmo propuesto son intermedios entre el algoritmo *Early classification* y *Enhanced K-Means*, la máxima pérdida de calidad fue de 2.55%. La experimentación con instancias reales fue similar a la experimentación de las instancias sintéticas, alcanzando en el mejor de los casos reducción de tiempo de 76.63% y reducción de calidad de 0.35%. Finalmente se

observó que el uso de esta propuesta se comporta mejor cuando el número de grupos es incrementado, se obtienen resultados similares a *Early classification* en cuanto al tiempo, pero un diferencia de calidad a comparación del algoritmo propuesto.

2.4.4. *Optimized big data K-Means clustering using MapReduce*

En [24], se presenta una optimización de la eficiencia del algoritmo de agrupación particional *K-Means*, esto es con el fin de obtener un alto rendimiento en la agrupación de los datos. La propuesta se plantea por un modelo de procesamiento llamado *MapReduce*, se utiliza el muestreo para eliminar la dependencia de iteración de *K-Means* y lograr un alto rendimiento.

Los detalles de la agrupación de *K-Means* en el marco de *MapReduce* se basan en trabajos de *MapReduce*. El primer trabajo, realiza particiones del conjunto de datos para ser procesadas de manera más rápida. La división de los datos se realiza usando *K*, probabilidad $p_x = \frac{1}{\varepsilon^2 * N}$, donde $\varepsilon \in (0, 1)$ y controlando el tamaño de la muestra.

El segundo trabajo de *MapReduce* se basa en agrupar cada uno de los archivos divididos. Se usan lo *K* grupos y se obtiene $2k^2$ centroides totales. Se combinan estos centroides con un solo reductor para obtener los *K* centroides que serán usados para obtener el resultado final de agrupamiento.

El tercer trabajo de *MapReduce* es realizar la agrupación, con los centroides obtenidos en el trabajo dos, se inicializan los *K* centroides al conjunto de datos iniciales, y después se obtiene la agrupación final.

Se realizaron pruebas con tres conjuntos de datos para evaluar el rendimiento del algoritmo. El primer conjunto de datos es sintético con una distribución gaussiana con 10,000 puntos tridimensionales. El segundo conjunto de datos *Bag of Words* consta de 2,351,710,420 puntos en tres dimensiones. El tercer conjunto de datos *Individual household electric power consumption* consta de 4,296,075,259 con nueve dimensiones.

Se hizo comparaciones del rendimiento del algoritmo *K-Means* optimizado, con el algoritmo *K-Means* estándar y *K-Means*|| referenciado en [25] usando *MapReduce*, y el algoritmo *K-Means*++ referenciado en [22] independiente. En las ejecuciones realizadas con el conjunto de datos sintético se usó valores de *K* en 20, 50 y 100. Se realizó un promedio de las iteraciones que tomó cada algoritmo hasta converger. Se realizó la aplicación de los trabajos

del algoritmo propuesto, y se hicieron comparaciones de tiempo entre los dos métodos propuestos con los algoritmos *K-Means, K-Means++* y *K-Means||*.

Después de realizar las pruebas y las comparaciones con los conjuntos de datos reales y sintéticos, se llegó a la conclusión de que el algoritmo optimizado es eficiente y funciona mejor a comparación de los algoritmos *K-Means, K-Means||* y *K-Means++*. Asimismo, la calidad de agrupación es tan buena como *K-Means* estándar.

En la Tabla 1, se presenta las características principales buscadas en los artículos estudiados del estado del arte.

Tabla 1. Características principales en los trabajos relacionados.

Variante	Etapa de mejora	Prueba frente a otros algoritmos	Describe los experimentos	Métrica de tiempo	Métrica de calidad	Instancias reales	Referencia
ElAgha initialization	Inicialización	*K-Means* estándar	Si	Indefinido	Distancia euclidiana	Si	[14]
D-K-Means	Inicialización	K-*Means* estándar	Si	Indefinido	Distancia euclidiana	Si	[15]
K++	Inicialización	*K-Means* estándar	Si	Indefinido	Distancia euclidiana	Si	[25]
HC	Clasificación	*K-Means* estándar	Si	Si	Sumatoria de error al cuadrado	Si	[1]
A-means	Clasificación	*K-Means* estándar, *Early clasification*	Si	Si	Índice de equidistancia	Si	[17]
Fahim	Clasificación	*K-Means* estándar, *CLARA algorithm*	Si	Si	Distancia euclidiana	Si	[19]
Proposed heuristic	Clasificación	*K-Means* estándar	Si	Si	Índice de equidistancia	No	[5]
N-Means	Clasificación	*K-Means* estándar	Si	Si	Sumatoria de error al cuadrado	No	[2]
Pattern reduction	Clasificación	*K-Means* estándar	Si	Si	Sumatoria de error al cuadrado	Si	[16]
Early stop heuristic	Convergencia	*K-Means* estándar	Si	Si	Sumatoria de error al cuadrado	Si	[21]
O-K-Means	Convergencia	*K-Means* estándar, *Fahim, Early classification*	Si	Si	Distancia euclidiana	Si	[4]
Efficient K-Means	Convergencia	*K-Means* estándar, *K-Medoid*	Si	Si	Distancia euclidiana	Si	[12]
A-Means, O-K-Means	Clasificación, convergencia	K-Means estándar, Early classification	Si	Si	Índice de equidistancia	Si	[8]
H-Kmeans	Inicialización, clasificación, convergencia	K-Means estándar	Si	Si	Distancia euclidiana	Si	[9]
Algoritmo hibrido	Clasificación, convergencia	K-Means estándar, Early classification, Enhanced K-Means	Si	Si	Distancia euclidiana	Si	[23]
K-Means optimizado	Inicialización, clasificación	K-Means estándar, K-Means ‖, K-Means++	Si	Si	Distancia euclidiana	Si	[24]

Capítulo 3

Estudio y clasificación de instancias

De acuerdo con la literatura especializada, se observó que existen dos tipos de instancias, las cuales son: sintéticas y reales. Las instancias sintéticas se caracterizan por ser generados para un entorno de prueba que necesite el usuario, además de contar con una distribución normal de datos para prácticas de agrupación manual.

Las instancias reales presentan características diferentes a una distribución normal, además de ser obtenidos de diferentes áreas como: la botánica, química, clima, economía, entre otras; estas instancias han sido puestos como datos abiertos por diferentes repositorios reconocidos. Como ejemplo en la Figura 2 se muestra la distribución que presentan las instancias sintéticas y las instancias reales existentes.

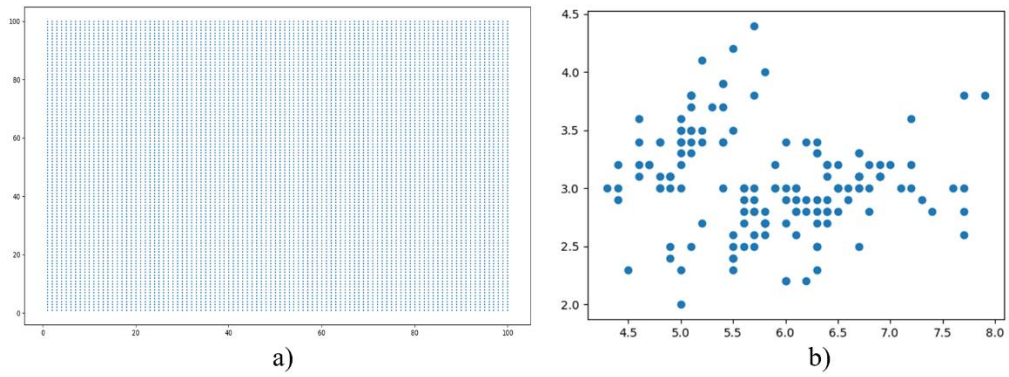


Figura 2. Distribución de instancias sintética y reales: a) Instancia sintética bidimensional con n = 10,000 distribuido en rango de 1–100; b) Instancia real *iris* con n = 150 y d = 2.

3.1 Instancias reconocidas

Mediante el análisis de publicaciones de la literatura relacionada con el algoritmo *K-Means*, se realizó una selección de diversos conjuntos de datos. En las publicaciones [14-16], [19], [24] y [28-32] se describen mejoras al algoritmo *K-Means* estándar. De esas publicaciones se seleccionaron los conjuntos de datos reales que utilizaron los autores. Además, se observó las distintas pruebas que realizaron los autores, para la comparación de las mejoras descritas.

Como resultado se recuperaron 37 conjuntos de datos, de las cuales 23 se presentan en un solo archivo y 14 de ellos en más de uno; para resolver este problema se realizó un pre-procesamiento, que consistió en dar un formato a todos los conjuntos de datos para implementar los casos experimentales. Se puede tener acceso a estos en repositorios reconocidos por la comunidad científica, tales como en [26] y [27]. Los conjuntos de datos seleccionados se describen a continuación:

3.1.1 Conjunto de datos *Iris*

Iris es un conjunto de datos real, puede ser la más conocida por su uso en el reconocimiento de patrones. Contiene datos de medidas obtenidas de tres tipos de plantas. El objetivo es agrupar los datos para diferenciar tipos de plantas. El conjunto de datos contiene 150 instancias y cuatro atributos [14-16] y [28-30].

3.1.2 Conjunto de datos *Appendicitis*

Appendicitis es un conjunto de datos real, describe las características médicas tomadas de 106 pacientes. El objetivo es conocer si el paciente tiene apendicitis o no. El conjunto de datos contiene 106 instancias y siete atributos [31].

3.1.3 Conjunto de datos *Newthyroid*

Newthyroid es un conjunto de datos real, describe las características sobre una enfermedad de la glándula tiroides. Contiene valores sobre la resina T3 o T3RU y diferentes tipos de hormonas. El objetivo es conocer si problemas en la glándula tiroides. El conjunto de datos tiene 215 instancias y cinco atributos [28] y [31].

3.1.4 Conjunto de datos *Glass identification*

Glass identification es un conjunto de datos real, contiene datos generados a partir de un sistema basado en reglas, mismo que busca clasificar tipos de vidrio encontrados en una escena de crimen, con él se puede identificar si el vidrio forma parte de una prueba. Este conjunto de datos contiene 214 instancias y nueve atributos [31] y [32].

3.1.5 Conjunto de datos *Bupa*

Bupa es un conjunto de datos real, contiene datos obtenidos de una evaluación de personas con posibles trastornos hepáticos. Contiene cinco valores sobre los trastornos y una variable más sobre la cantidad de bebidas alcohólicas ingeridas al día. El conjunto de datos contiene 345 instancias y seis atributos [30].

3.1.4 Conjunto de datos *Wine*

Wine es un conjunto de datos real, contiene datos de un análisis químico de vinos que fueron cultivados en Italia pero de distintos cultivares. El conjunto de datos contiene valores del ácido málico, magnesio entre otras. El objetivo es diferenciar los vinos de otros mediante sus características. El conjunto de datos tiene 179 instancias y 13 atributos [28] y [29].

3.1.7 Conjunto de datos *Ecoli*

Ecoli es un conjunto de datos real, contiene datos de un estudio sobre una bacteria. Se busca la localización de las proteínas empleando las medidas sobre la célula (citoplasma, membrana interna, peris plasma, membrana externa, lipoproteína de membrana interna y externa). El conjunto de datos contiene 336 instancias y siete atributos [32].

3.1.8 Conjunto de datos *Balance scale*

Balance scale es un conjunto de datos real. Este conjunto de datos contiene valores para tener una balanza que se caracteriza por el peso y distancia que pueda tener en ambos lados de la balanza. El objetivo es conocer hacia donde se inclina la balanza. Este conjunto de datos contiene 625 instancias y cuatro atributos [29] y [31].

3.1.9 Conjunto de datos *Heart disease*

Heart disease es un conjunto de datos real, contiene un subconjunto de 74 atributos. En él se muestra 13 atributos que se han usado mayormente para la búsqueda de una enfermedad cardíaca en el paciente. Sus valores para ello van de cero a cuatro. El conjunto de datos contiene 294 instancias y 13 atributos [32].

3.1.10 Conjunto de datos *Cleveland*

Cleveland es un conjunto de datos real, es un subconjunto de un conjunto de datos sobre enfermedades cardiacas del centro médico *V. A. Long Beach* y la fundación clínica Cleveland. Consiste en detectar la presencia de una enfermedad cardíaca en el paciente. El conjunto de datos contiene 304 instancias y 13 atributos [31].

3.1.11 Conjunto de datos *Steel plates faults*

Steel plates faults es un conjunto de datos real, contiene datos sobre medidas de placas de acero. En este se clasifican 7 diferentes tipos de fallas, y tiene el objetivo de ser usado por aprendizaje automático para el reconocimiento de patrones. El conjunto de datos contiene 194 instancias y 27 atributos [32].

3.1.12 Conjunto de datos *Pima*

Pima es un conjunto de datos real, es una versión desequilibrada del conjunto de datos *Pima indias*. Este tiene clases renombradas como positivas y negativas. El conjunto de datos contiene 768 instancias y 8 atributos [29].

3.1.13 Conjunto de datos *Pima indians*

Pima indians es un conjunto de datos real, contiene datos obtenidos del instituto nacional de diabetes y enfermedades digestivas y renales. En él se busca conocer si la persona padece diabetes o no. Los pacientes son mujeres de al menos 21 años de origen *Pima*. El conjunto de datos contiene 768 instancias y ocho atributos [32].

3.1.14 Conjunto de datos *Breast cancer wisconsin*

Breast cancer wisconsin es un conjunto de datos real, contiene datos agrupados de informes que se realizaron en una clínica de manera periódica que va desde enero de 1989 a noviembre de 1991. Se busca evaluar si un tumor es maligno o benigno. El conjunto de datos contiene 699 instancias y nueve atributos [29] y [32].

3.1.15 Conjunto de datos *Concrete data*

Concrete data es un conjunto de datos real, contiene datos de la cantidad de ingredientes y edad que puede tener una combinación de concreto. El objetivo es conocer mediante las características si el concreto es resistente o no. El conjunto de datos contiene 1,030 instancias y nueve atributos [32].

3.1.16 Conjunto de datos *Cloud*

Cloud es un conjunto de datos real, contiene datos que forman vectores obtenidos de dos imágenes AVHRR. Cada imagen se divide en superpíxeles de 16*16 y en cada superpíxel se calcula un conjunto de parámetros. El conjunto de datos contiene 1,024 instancias y 10 atributos [32].

3.1.17 Conjunto de datos *Spectf*

Spectf es un conjunto de datos real, describe el diagnóstico de imágenes cardíacas de tomografía computarizada por emisión de protón único. Contiene valores extraídos de las imágenes computarizadas para definir si un paciente se categoriza como normal o anormal. El conjunto de datos contiene 267 instancias y 44 atributos [31] y [32].

3.1.18 Conjunto de datos *Yeast*

Yeast es un conjunto de datos real, contiene una predicción de los sitios de localización celular de proteínas. Este es un subconjunto de uno más grande, mayormente usado para la clasificación basado en reglas. Este conjunto de datos contiene 1,484 instancias y ocho atributos [32].

3.1.19 Conjunto de datos *Vehicle silhouettes*

Vehicle silhouettes es un conjunto de datos real, es usada para la clasificación de una silueta que puede formar parte de uno de cuatro tipos de carros. El objetivo es conocer desde qué ángulo se observa un vehículo, dependiendo de la silueta del mismo. El conjunto de datos contiene 846 instancias y 18 atributos [29] y [32].

3.1.20 Conjunto de datos *Abalone*

Abalone es un conjunto de datos real, contiene datos de mediciones físicas. El objetivo es determinar las edades de los abulones. La edad del abulón se determina cortando la concha a través del cono, tiñéndola y contando el número de anillos a través de un microscopio. El conjunto de datos contiene 4,177 instancias y siete atributos [19].

3.1.21 Conjunto de datos *Image segmentation*

Image segmentation es un subconjunto de una base de datos real de segmentación de imágenes. Estos datos fueron extraídos de la base de datos de siete imágenes de exteriores. Estas imágenes se segmentaron a mano para crear una clasificación para cada píxel. El conjunto de datos contiene 2,310 instancias y 19 atributos [32].

3.1.22 Conjunto de datos *Page blocks clasification*

Este conjunto de datos real contiene ejemplos sobre la clasificación de bloques que componen una página de un documento que ha sido detectado por un proceso de segmentación. Los datos provienen de 54 documentos distintos y cada observación se refiere a un bloque. El conjunto de datos contiene 5,473 instancias y 10 atributos [32].

3.1.23 Conjunto de datos *Wine quality*

Wine quality es un conjunto de datos real, contiene datos que relacionan dos tipos de vino: rojo y blanco, mismos que son variantes del "*Vinho* verde" portugués. Las características que describen el conjunto de datos va desde la acidez volátil, ácido cítrico, densidad, pH, entre otros. El conjunto de datos contiene 6497 instancias y 12 atributos [32].

3.1.24 Conjunto de datos Parkinsons

Parkinsons es un conjunto de datos real, está compuesto por una serie de medidas biomédicas de la voz de 31 personas, 23 de ellas con *Parkinson*. Cada atributo es una medida de voz concreta y cada fila corresponde a una de 195 grabaciones. El objetivo es identificar personas sanas de las que padecen. El conjunto de datos contiene 5,875 instancias y 21 atributos [32].

3.1.25 Conjunto de datos *Wall following*

Wall fllowing es un conjunto de datos real, está compuesto por tres conjuntos de datos diferentes. Los datos fueron recopilados de 24 sensores de ultrasonidos, distancias simplificadas delantera, trasera, izquierda y derecha. El conjunto de datos contiene 5,456 instancias y 27 atributos [32].

3.1.26 Conjunto de datos *Pen digits*

Pen digits es un conjunto de datos real, contiene 250 muestras de 44 escritores. Las muestras escritas por 30 escritores se utilizan para el entrenamiento, la validación cruzada y

la prueba dependiente del escritor, y los dígitos escritos por los otros 14 se utilizan para la prueba independiente del escritor. El conjunto de datos contiene 10,992 instancias y 16 atributos [32].

3.1.27 Conjunto de datos *Magic gamma telescope*

Magic gamma telescope es un conjunto de datos real, contiene datos generados de MC para simular el registro de partículas gamma de alta energía en un telescopio gamma *Cherenkov* atmosférico basado en tierra utilizando la técnica de imagen. El conjunto de datos contiene 19,020 instancias y 10 atributos [32].

3.1.28 Conjunto de datos *Isolet*

Isolet es un conjunto de datos real, contiene datos generados de 150 sujetos que pronunciaron dos veces el nombre de cada letra del alfabeto. El conjunto de datos contiene 7,797 instancias y 29 atributos [32].

3.1.29 Conjunto de datos *Landsat satellite*

Landsat satellite es un conjunto de datos real, contiene valores multiespectrales de los píxeles en 3*3 formados de una imagen de satélite, y la clasificación asociada al píxel central de cada forma. El objetivo es predecir esta clasificación, dados los valores multiespectrales. El conjunto de datos contiene 6,435 instancias y 36 atributos [32].

3.1.30 Conjunto de datos *Letter recognition*

Letter recognition es un conjunto de datos real, es usada para identificar letras desplegadas en pantallas de píxeles rectangulares en blanco y negro. Este conjunto de datos contiene 20,000 instancias y 16 atributos [19] y [32].

3.1.31 Conjunto de datos *Optical digits*

Optical digits es un conjunto de datos real, contiene valores de un pre procesamiento para extracción de mapas de bits, normalizados de dígitos manuscritos de un formulario pre impreso. De un total de 43 personas, 30 pertenecen al conjunto de entrenamiento y otras 13 al conjunto de prueba. El conjunto de prueba contiene 5,620 instancias y 64 atributos [32].

3.1.32 Conjunto de datos *Shuttle*

Shuttle es un conjunto de datos real, contiene ejemplos de un conjunto de datos que estaba en orden temporal, el cual podría ser relevante en la clasificación. Este ejemplo se creó

seleccionando algunos datos de manera aleatoria. 80% de los datos pertenecen a la clase uno. El conjunto de datos contiene 58,000 instancias y nueve atributos [32].

3.1.33 Conjunto de datos *Person activitis*

Person activitis es un conjunto de datos real, contiene registros de sensores que usaban personas en los tobillos, cinturón y pecho. La finalidad de estos sensores era registrar valores de localización para conocer qué actividad estaban realizando las personas. El conjunto de datos contiene 164,820 instancias y cuatro atributos [32].

3.1.34 Conjunto de datos *Musk clean*

Musk clean es un conjunto de datos real, contiene la descripción de 102 moléculas, de las cuales 39 son consideradas como almizcles y 63 como no almizcles. El objetivo de este conjunto es predecir si las nuevas moléculas serán almizcles o no. El conjunto de datos contiene 6,598 instancias y 166 atributos [32].

3.1.35 Conjunto de datos *Corel image features*

Corel es un conjunto de datos real, contiene datos que son características de imágenes extraídas de una colección de imágenes de Corel. En este conjunto se evalúan características basadas en el histograma de color, disposición de histograma de color, momentos de color y concurrencia. El conjunto de datos contiene 66,616 instancias y 89 atributos [32].

3.1.36 Conjunto de datos *Bag of words*

Bag of words es un conjunto de datos real, contiene cinco colecciones de texto en forma de bolsas de palabras. En este conjunto de datos se utilizó una partición de ella, al observar que son diversas bolsas de palabras de diferentes documentos se optó por utilizar una bolsa de palabras. El conjunto de datos que se utilizó contiene 3,710,420 instancias y 3 atributos [24].

3.1.37 Conjunto de datos *Covertype*

Covertype es un conjunto de datos real, contiene valores sobre tipo de cubierta forestal a partir de variables cartográficas. El área de estudio incluye cuatro áreas silvestres situadas en el Bosque Nacional *Roosevelt* del norte de Colorado. El conjunto de datos contiene 581,012 instancias y 50 atributos [32].

3.2 Clasificación

El propósito de realizar una clasificación entre estas instancias es: definir un límite para instancias pequeñas y para instancias grandes. La definición de ese límite se puede observar en la complejidad generada de *n*d*. En la Tabla 1, se presentan las instancias reales recopiladas, donde *n* representa el número de objetos, *d* representa el número de dimensiones, *n*d* representa el valor de complejidad y referencia contiene las publicaciones que atribuyen el uso de estas instancias. Estas instancias se caracterizan por contar con valor menor a 100,000 dentro de la complejidad de *n*d*.

Tabla 2. Concentrado de instancias reales pequeñas.

ID	Instancia	*n*	*d*	*n*d*	Referencia
1	*Iris*	150	4	600	[14], [15], [16], [28], [29], [30]
2	*Appendicitis*	106	7	742	[31]
3	*Newthyroid*	215	5	1,075	[28], [31]
4	*Glass identification*	214	9	1,926	[31], [32]
5	*Bupa*	345	6	2,070	[30]
6	*wine*	179	13	2,327	[28], [29]
7	*Ecoli*	336	7	2,352	[32]
8	*Balance scale*	625	4	2,500	[29], [31]
9	*Heart disease*	294	13	3,822	[32]
10	*Cleveland*	304	13	3,952	[31]
11	*Steel plates faults*	194	27	5,238	[32]
12	*Pima*	768	8	6,144	[29]
13	*Pima indians*	768	8	6,144	[32]
14	*Breast cancer wisconsin*	699	9	6,291	[29], [32]
15	*Concrete data*	1,030	9	9,270	[32]
16	*Cloud*	1,024	10	10,240	[32]
17	*Spectf*	267	44	11,748	[31], [32]
18	*yeast*	1,484	8	11,872	[32]
19	*Vehicle silhouettes*	846	18	15,228	[29], [32]
20	*Abalone*	4,177	7	29,239	[19]
21	*Image segmentation*	2,310	19	43,890	[32]
22	*Page blocks clasification*	5,473	10	54,730	[32]
23	*Wine quality*	6,497	12	77,964	[32]

Como se muestra en la Tabla 1, se describen los valores de 23 instancias de prueba clasificadas como pequeñas. En la Tabla 2 se describen 14 instancias más, las cuales se clasificaron como instancias grandes. A diferencia de las instancias pequeñas, el valor de *n*d* es mayor a 100,000.

Tabla 3. Concentrado de instancias reales grandes.

ID	Instancia	*n*	*d*	*n*d*	Referencia
24	*Parkinsons*	5,875	21	123,375	[32]
25	*Wall following*	5,456	27	147,312	[32]
26	*Pen digits*	10,992	16	175,872	[32]
27	*Magic gamma telescope*	19,020	10	190,200	[32]
28	*Isolet*	7,797	29	226,113	[32]
29	*Landsat satellite*	6,435	36	231,660	[32]
30	*Letter recognition*	20,000	16	320,000	[19], [32]
31	*Optical digits*	5,620	64	359,680	[32]
32	*Shuttle*	58,000	9	522,000	[32]
33	*Person Activitis*	164,860	4	659,440	[32]
34	*Musk clean*	6,598	166	1,095,268	[32]
35	*Corel image features*	66,616	89	5,928,824	[32]
36	*Bag of words*	3,710,420	3	11,131,260	[24]
37	*Covertype*	581,012	50	29,050,600	[32]

Capítulo 4

Experimentación y análisis de resultados

El propósito de la experimentación que se describe en este capítulo, fue comprobar en qué condiciones una variante del algoritmo *K-Means* obtiene menor tiempo de ejecución y menor pérdida de calidad. Para realizar esta experimentación, se optó por implementar dos mejoras, ya existentes, del algoritmo *K-Means*, como base para obtener un índice de referencia de la calidad de agrupamiento y del tiempo de procesamiento. Estas mejoras fueron *Fahim* [19] y *O-K-Means* [4]. La selección de dos variantes se debe a que se observó que la variante *Fahim* es mayormente reconocido e implementado frente a diversas mejoras para comparar resultados. Por otro lado, *O-K-Means* es una variante relevante la cual se observó que frente a diversas pruebas genera un menor costo computacional al acelerar la etapa de convergencia del algoritmo *K-Means.*

4.1 Selección de variantes de *K-Means*

Las variantes que se implementaron, se tomaron del estudio realizado del estado del arte, las cuales se describen a continuación:

Fahim, descrita en [19], se caracteriza por comparar las distancias de los objetos a sus centroides en cada iteración. Este procedimiento, se realiza por medio de una nueva función

que consiste en almacenar las distancias de los objetos a sus centroides. Posteriormente, las compara con las distancias calculadas en una iteración anterior. Además, tiene una condición que permite comparar los valores de distancia de cada nueva iteración con las sucesivas iteraciones. De este modo, si la distancia obtenida en la siguiente iteración es menor que la distancia calculada en la iteración anterior, el objeto conserva su grupo y se cancelan los cálculos de distancias del objeto a su centroide en las próximas iteraciones.

La variante *O-K-Means*, descrita en [4], se caracteriza por mejorar la etapa de convergencia del algoritmo. Esta variante, agrega un umbral de convergencia, el cual se obtuvo observando la disminución del valor de la función objetivo a través de diferentes iteraciones. Dicho umbral, se compara con el porcentaje de objetos que cambian de grupo. Cuando el porcentaje es menor al umbral definido, el algoritmo converge.

4.2 Diseño experimental

La implementación del algoritmo *K-Means* y sus variantes, se realizó en lenguaje *C*. Para implementarlo, se tomó como referencia el código del algoritmo *K-Means* estándar. Con base en el diseño experimental, se realizaron 222 ejecuciones, tomando como base 37 instancias, las cuales se resolvieron con tres algoritmos diferentes; el algoritmo *K-Means* estándar, la variante *Fahim* y *O-K-Means*; y con dos valores distintos de *k*, uno de ellos fue con k=5 y el otro con k=10. Esto con el propósito de ver si hay algún cambio en el dominio de solución de instancias cuando se duplica el valor de *K*.

Es conveniente mencionar que, para la solución de cada instancia, se usaron los mismos centroides iniciales al momento de ejecutar cada variante del algoritmo *K-Means*, estos centroides fueron seleccionados de manera aleatoria partiendo de los datos de cada instancia.

El diseño experimental se desarrolló en dos etapas, de la siguiente manera:

1) En el experimento A, se resolvieron todas las instancias usando un valor de k=5, para observar la ventaja que presenta alguna de las variantes en cuanto a pérdida de calidad o reducción de tiempo.
2) En el experimento B, se resolvieron todas las instancias usando un valor de k=10 con la finalidad de observar los cambios en las variantes al duplicar los grupos.

Es posible replicar este diseño experimental porque se generaron carpetas de ejecución para su implementación en distintos equipos. Además, para obtener los mismos resultados, es necesario el uso del compilador "*gcc*" versión 7.4.0 para ejecutar código en lenguaje *C*. Es conveniente mencionar, que los experimentos que se presentan en la Sección 4.3, se realizaron con un equipo de cómputo. Las características del equipo son: procesador Intel Core i3-3220 CPU @ 3.30 GHz, 12 GB DDR3 de RAM, tarjeta gráfica NVIDIA *GeForce* GT 630 a 2 GB y sistema operativo Windows 10 Pro 64 bits.

4.3 Dominio de las variantes en el tiempo

Mediante el análisis de los resultados obtenidos, se realizó una búsqueda de patrones de solución. Para ello, se implementó un índice comparativo entre el algoritmo *K-Means* estándar y sus dos variantes, con el fin de determinar el porcentaje de reducción de tiempo con cada variante, lo cual permitirá identificar aquellas instancias que se resolvieron en el menor tiempo posible. La métrica de evaluación se presenta en la Ecuación 1:

$$T = 100 - \frac{tv*100}{tk} \qquad (1)$$

- *T* representa la diferencia en porcentaje de la variante respecto a *K-Means*.
- *tk* representa el tiempo de ejecución del algoritmo *K-Means* estándar.
- *tv* representa el tiempo de ejecución de la variante utilizada.

Los resultados de porcentaje pueden tomar valores negativos o positivos. Un valor positivo representa una reducción de tiempo en la solución de una instancia dada. Un valor negativo representa un aumento de tiempo en la solución de una instancia dada. Nótese que, lo que se busca es una diferencia del tiempo de solución del algoritmo *K-Means* estándar [33].

Tomando en cuenta la cantidad de experimentos, se realizó un cálculo del porcentaje de las instancias analizadas. La Figura 3, muestra el porcentaje obtenido del dominio de las variantes en el tiempo en cada uno de los experimentos. Se puede observar que la variante *O-K-Means* destaca en el experimento A; mientras que en el experimento B, la variante *Fahim* obtuvo mayor dominio sobre las instancias.

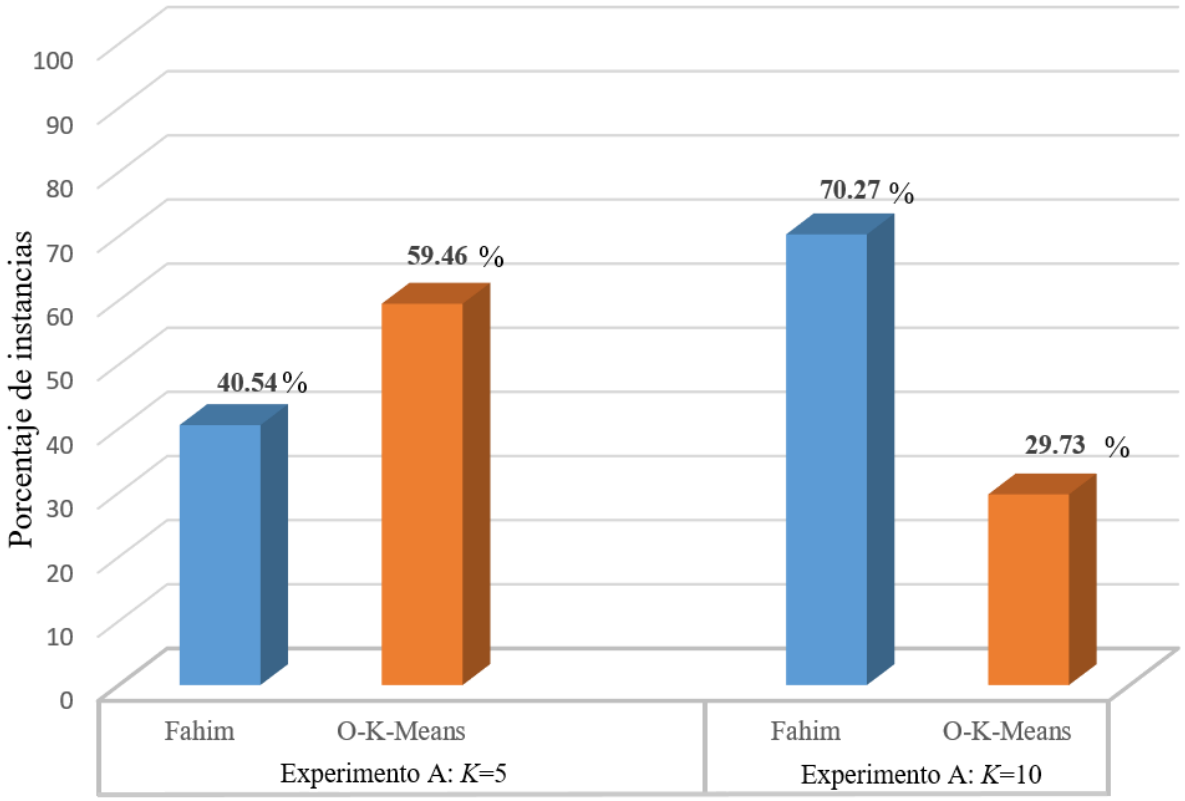


Figura 3. Porcentaje de dominio de las variantes *Fahim* y *O-K-Means* con la métrica de tiempo. Experimentos A y B.

En la Figura 3, se muestra el porcentaje de dominio de cada variante dentro de las instancias de prueba aplicando la métrica de tiempo. El total de 37 instancias representan el 100%. En el experimento A, se observa una ventaja de *Fahim* sobre 15 instancias (40.54%), mientras que *O-K-Means* obtuvo ventaja sobre 22 instancias (59.46%). En el experimento B, se observa una ventaja de *Fahim* sobre 26 instancias (70.27%), mientras que *O-K-Means* obtuvo ventaja en 11 instancias (29.73%).

4.4 Dominio de variantes en calidad

Con los mismos datos utilizados en la sección anterior, se realizó un filtrado similar al de la búsqueda de patrones en el tiempo. Para realizarlo, se implementó un índice comparativo, que demostró la diferencia en la calidad de solución que tuvieron las variantes, respecto al algoritmo *K-Means* estándar. Cabe mencionar, que la calidad de la solución es la suma de los cuadrados de cada objeto a su centroide. La mayoría de los valores obtenidos fueron términos negativos y muy pocos fueron mayores o iguales a cero, esto se explicará a continuación. La métrica de evaluación se denota con la Ecuación 2:

$$V = \frac{Zk - Zv}{zk} * 100 \quad (2)$$

- *V* representa la diferencia en porcentaje de la variante respecto a *K-Means.*
- *Zk* representa el valor de la función objetivo del algoritmo *K-Means* estándar.
- *Zv* representa el valor de la función objetivo de cada una de las variantes aplicadas.

Los resultados de porcentaje pueden tomar valores negativos o positivos. Un valor positivo representa una mejora en la calidad de solución con una instancia dada. Un valor negativo representa una pérdida en la calidad de solución con una instancia dada. Nótese que, lo que se busca es una diferencia del total de la función objetivo del algoritmo *K-Means* estándar.

Se realizó un cálculo del porcentaje de las instancias utilizadas, similar al que se realizó con la búsqueda de patrones de reducción de tiempo. La Figura 4, muestra el porcentaje de dominio que presenta cada una de las variantes sobre las instancias resueltas. En ellas se destacó la que obtuvo menor pérdida de calidad.

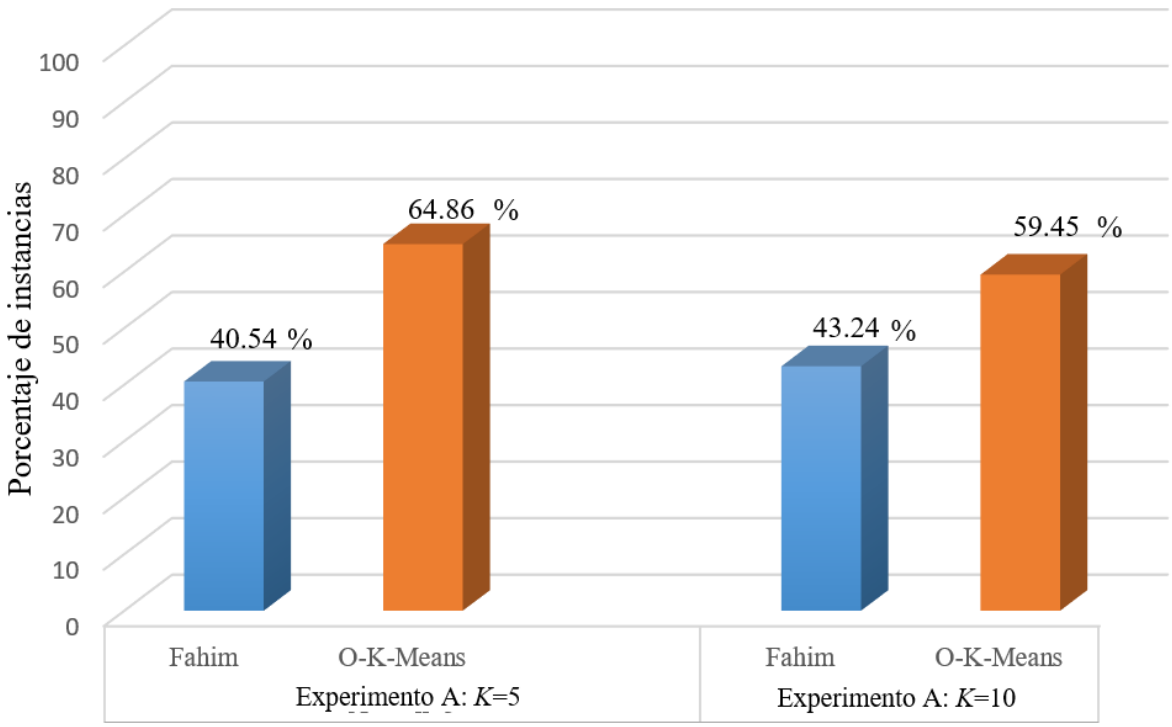


Figura 4. Porcentaje de dominio de las variantes *Fahim* y *O-K-Means* con la métrica de calidad. Experimentos A y B.

Como se observa en la Figura 4, se muestra el porcentaje del total de las instancias (37) en las que *Fahim* tuvo menor dominio con la métrica de calidad. Éstas fueron 15 instancias (40.54%) en el experimento A, y 16 instancias (43.24%) en el experimento B. En cambio, la variante *O-K-Means*, obtuvo mayor dominio con la métrica de calidad, representada por 24 instancias (64.86%) en el experimento A, y por 22 instancias (59.45%) en el experimento B.

4.5 Desempeño de variantes con instancias pequeñas

Con los resultados obtenidos en los apartados anteriores, se identificó el desempeño de las variantes en cada instancia, con lo cual, se logró identificar cuál fue la variante dominante en tiempo y en calidad. Una característica importante, es que, con instancias pequeñas, se presentan diferencias en los valores de los objetos y dimensiones.

El propósito de resolver instancias pequeñas es para conocer qué variante reduce el tiempo de solución y qué variante mejora la calidad de agrupación. Este conocimiento es útil para destacar si hay diferencia en el dominio de solución de instancias grandes. A continuación, se muestra una clasificación de instancias pequeñas que fueron agrupadas con las variantes

Fahim y *O-K-Means*. En la Tabla 3, se muestran los porcentajes de dominio respecto a la reducción de tiempo o pérdida de calidad. Para conocer los resultados por cada conjunto de datos resuelto, ir al Anexo A.

Tabla 4. Desempeño de las variantes *Fahim* y O-K-*Means* con instancias pequeñas.

ID	Instancia	*K*	Tiempo	Tiempo (%)	Calidad	Calidad (%)
1	*Iris*	5	*Fahim*	53.95	*O-K-Means*	0
		10	*Fahim*	72.07	*O-K-Means*	0
2	*Appendicitis*	5	*Fahim*	54.63	*O-K-Means*	0
		10	*Fahim*	61.43	*O-K-Means*	0
3	*New thyroid*	5	*Fahim*	55.78	*O-K-Means*	0
		10	*Fahim*	51.64	*O-K-Means*	0
4	*Glass identification*	5	*O-K-Means*	0.43	*Fahim*	0.33
		10	*Fahim*	32.30	*O-K-Means*	0
5	*Bupa*	5	*Fahim*	21.05	*O-K-Means*	0
		10	*O-K-Means*	49.25	*O-K-Means*	-0.15
6	*Wine*	5	*Fahim*	76.20	*O-K-Means*	0
		10	*Fahim*	62.45	*O-K-Means*	-5.71
7	*Ecoli*	5	*O-K-Means*	14.53	*O-K-Means*	0
		10	*Fahim*	61.94	*Fahim*	-0.02
8	*Balance scale*	5	*O-K-Means*	12.43	*O-K-Means*	-0.008
		10	*Fahim*	52.26	*O-K-Means*	0
9	*Heart disease*	5	*Fahim*	34.69	*O-K-Means*	0.03
		10	*Fahim*	68.96	*O-K-Means*	-0.35
10	*Cleveland*	5	*Fahim*	49.53	*Fahim*	0.28
		10	*Fahim*	59.28	*O-K-Means*	-0.07
11	*Steel plates faults*	5	*Fahim*	26.20	*O-K-Means*	0.29
		10	*Fahim*	64.91	*O-K-Means*	0.31
12	*Pima*	5	*Fahim*	45.40	*O-K-Means*	-0.007
		10	*Fahim*	38.13	*Fahim*	0.05
13	*Pima indians*	5	*O-K-Means*	44.24	*O-K-Means*	-0.01
		10	*Fahim*	45.99	*O-K-Means*	-0.08
14	*Breast cancer wisconsin*	5	*O-K-Means*	47.82	*O-K-Means*	-0.44
		10	*Fahim*	71.98	*O-K-Means*	-0.05
15	*Concrete data*	5	*Fahim*	74.31	*O-K-Means*	-0.01
		10	*Fahim*	76.76	*O-K-Means*	-0.15
16	*Cloud*	5	*O-K-Means*	47.49	*O-K-Means*	1.86
		10	*Fahim*	51.003	*O-K-Means*	-0.006
17	*Spectf*	5	*Fahim*	9.77	*O-K-Means*	0
		10	*Fahim*	30.32	*Fahim*	0.01
18	*Yeast*	5	*O-K-Means*	37.19	*Fahim*	0.02
		10	*O-K-Means*	73.37	*Fahim*	0.51
19	*Vehicle silhouettes*	5	*Fahim*	40.51	*O-K-Means*	-0.21
		10	*Fahim*	49.21	*O-K-Means*	-0.18
20	*Abalone*	5	*O-K-Means*	34.43	*Fahim*	0
		10	*Fahim*	79.14	*Fahim*	-0.88
21	*Steel plates faults*	5	*Fahim*	26.20	*O-K-Means*	0.29
		10	*Fahim*	64.91	*O-K-Means*	0.31
22	*Vehicle silhouttes*	5	*Fahim*	40.51	*O-K-Means*	-0.21
		10	*Fahim*	49.21	*O-K-Means*	-0.18
23	*Wine quality*	5	*Fahim*	85.32	*O-K-Means*	-0.52
		10	*Fahim*	47.22	*O-K-Means*	0.06

Como se muestra en la Tabla 3, la variante *Fahim* demostró mejor desempeño en cuanto a reducción de tiempo con 78% sobre el total de instancias. En algunos resultados es conveniente destacar que se redujo el tiempo de solución de la instancia *Wine quality* hasta un 85.32% con respecto al algoritmo *K-Means* estándar. Los resultados marcados con color crema, fueron destacables frente a los demás. Para conocer los valores obtenidos en el tiempo por la variante *Fahim* ver anexo A, para conocer los valores obtenidos en el tiempo por la variante *O-K-Means* ver anexo B.

Por el contrario, la variante *O-K-Means*, presentó mejor desempeño al obtener menor pérdida de calidad con 80% sobre el total de instancias. Para conocer los valores obtenidos en la calidad de agrupación por la variante *Fahim* ver anexo A, para conocer los valores obtenidos en la calidad de agrupación por la variante *O-K-Means* ver anexo B.

Finalmente, con los resultados obtenidos al aplicar ambas variantes, se concluye que cuando se desea reducir el tiempo al agrupar instancias pequeñas, es conveniente aplicar la variante *Fahim*. En cambio, la variante *O-K-Means,* se aplica cuando se espera obtener la menor pérdida de calidad posible.

4.6 Desempeño de variantes con instancias grandes

En este apartado, se presenta una clasificación con instancias grandes, similar a la que se presentó en la Sección 4.5. Estas instancias, presentan una estructura similar a las de instancias pequeñas. En la Tabla 4, se muestra el desempeño de las variantes *Fahim* y *O-K-Means* con instancias grandes.

Tabla 5. Desempeño de las variantes *Fahim* y *O-K-Means* con instancias grandes.

ID	Instancia	*K*	Tiempo	Tiempo (%)	Calidad	Calidad (%)
24	*Parkinson*	5	*O-K-Means*	71.37	*Fahim*	-0.28
		10	*Fahim*	55.47	*O-K-Means*	-0.20
25	*Wall following*	5	*O-K-Means*	37.42	*Fahim*	0.001
		10	*O-K-Means*	63.06	*Fahim*	-0.02
26	*Pen digits*	5	*O-K-Means*	63.14	*Fahim*	0.0001
		10	*O-K-Means*	81.003	*Fahim*	0.0002
27	*Magic gamma telescope*	5	*Fahim*	16.03	*Fahim*	-0.001
		10	*Fahim*	62.76	*Fahim*	0.71
28	*Isolet*	5	*O-K-Means*	68.91	*Fahim*	-0.004
		10	*O-K-Means*	51.80	*Fahim*	-0.0008
29	*Landsat satellite*	5	*O-K-Means*	62.06	*Fahim*	-0.02
		10	*O-K-Means*	74.33	*Fahim*	-0.82
30	*Letter recognition*	5	*O-K-Means*	53.93	*Fahim*	-0.0005
		10	*Fahim*	32.30	*O-K-Means*	-0.12
31	*Optical digits*	5	*O-K-Means*	25.57	*Fahim*	-0.006
		10	*Fahim*	55.17	*O-K-Means*	-0.002
32	*Shuttle*	5	*O-K-Means*	36.67	*O-K-Means*	0.06
		10	*O-K-Means*	50.64	*Fahim*	0.57
33	*Person activits*	5	*Fahim*	71.55	*O-K-Means*	0
		10	*O-K-Means*	78.74	*Fahim*	0.04
34	*Musk clean*	5	*O-K-Means*	61.40	*Fahim*	0
		10	*Fahim*	52.82	*Fahim*	-0.0006
35	*Corel image features*	5	*O-K-Means*	69.23	*O-K-Means*	0.21
		10	*O-K-Means*	69.80	*O-K-Means*	0.03
36	*The bag of words*	5	*O-K-Means*	74.97	*O-K-Means*	-0.02
		10	*O-K-Means*	83.98	*Fahim*	0.11
37	*Covertype*	5	*O-K-Means*	90.13	*O-K-Means*	0.09
		10	*O-K-Means*	93.57	*Fahim*	-0.903

Como se muestra en la Tabla 4, al aplicar la variante *O-K-Means*, se obtuvo un mejor desempeño porque se logró menor tiempo de ejecución con 75% sobre el total de instancias. Por el contrario, al aplicar la variante *Fahim*, se demostró un mejor desempeño porque se pudo reducir la pérdida de calidad con 68% sobre el total de instancias.

Finalmente, con los resultados obtenidos, se concluye que cuando es necesario agrupar instancias grandes y reducir el tiempo de ejecución es conveniente aplicar la variante *O-K-Means*. En el caso de que se busque reducir la pérdida de calidad para agrupar instancias grandes es conveniente aplicar la variante *Fahim*, del algoritmo *K-Means*.

4.7 Comparación de dominio entre variantes

En este apartado, se presenta la comparación de la variante *Fahim* y *O-K-Means* al resolver una instancia grande. El propósito es, conocer el porcentaje de dominio de una variante sobre otra. En la Tabla 5, se muestra la diferencia de porcentaje de dominio en el tiempo de las variantes *Fahim* y *O-K-Means* con las respectivas instancias grandes. La columna Tf (%) representa el porcentaje de dominio de la variante *Fahim* sobre el tiempo, la columna To (%) representa el porcentaje de dominio de la variante *O-K-Means* sobre el tiempo, la última columna representa la diferencia en porcentaje de los resultados obtenidos por la variante *Fahim* y *O-K-Means.* Los datos marcados con color crema muestran los mejores resultados.

Tabla 6. Diferencia de dominio en el tiempo entre la variante *Fahim* y *O-K-Means*.

ID	Instancia	K	Tf (%)	To(%)	Variante dominante (%)
24	*Parkinson*	5	61.05	71.37	10.32
		10	55.47	39.93	15.54
25	*Wall following*	5	26.18	37.42	11.24
		10	27.49	63.06	35.57
26	*Pen digits*	5	34.51	63.14	28.63
		10	57.75	81.00	23.25
27	*Magic gamma telescope*	5	16.03	8.62	7.41
		10	62.76	44.31	18.45
28	*Isolet*	5	21.20	68.91	47.71
		10	41.98	51.80	9.82
29	*Landsat satellite*	5	20.93	62.06	41.13
		10	55.39	74.33	18.94
30	*Letter recognition*	5	25.71	53.93	28.22
		10	52.16	31.91	20.25
31	*Optical digits*	5	21.49	25.57	4.08
		10	55.17	16.39	38.78
32	*Shuttle*	5	25.82	36.67	10.85
		10	46.20	50.64	4.44
33	*Person activits*	5	71.55	-0.61	72.16
		10	73.75	78.74	4.99
34	*Musk clean*	5	46.98	61.40	14.42
		10	52.82	35.70	17.12
35	*Corel image features*	5	-4.35	69.23	73.58
		10	41.29	69.80	28.51
36	*The bag of words*	5	-1.44	74.97	76.41
		10	-60.25	83.98	144.23
37	*Covertype*	5	39.41	90.13	50.72
		10	83.57	93.57	10.00

Como se muestra en la Tabla 5, se observa que es conveniente usar la variante *O-K-Means* para resolver instancias grandes en menor tiempo de ejecución. La variante *O-K-Means*, obtuvo una diferencia de porcentaje respecto a la variante *Fahim* de hasta 144.23% en el mejor de los casos. La variante *Fahim* obtuvo dominio frente a la variante *O-K-Means* en pocos casos, en el mejor de ellos alcanzó un dominio de hasta 72.16%. En la Tabla 6, se muestra la diferencia de porcentaje de dominio en la calidad de solución de las variantes *Fahim* y *O-K-Means* con las respectivas instancias grandes. Los datos sombreados muestran los mejores resultados.

Tabla 7. Diferencia de dominio en la calidad entre la variante *Fahim* y *O-K-Means*.

ID	Instancia	K	Zf (%)	Zo(%)	Variante dominante (%)
24	*Parkinson*	5	-0.28	-2.73	2.45
		10	-0.49	-0.20	0.29
25	*Wall following*	5	0.001	-0.004	0.005
		10	-0.02	-0.04	0.02
26	*Pen digits*	5	0.0001	-0.01	0.0101
		10	0.0002	-1.67	1.6702
27	*Magic gamma telescope*	5	-0.001	-12.52	12.519
		10	0.71	-5.58	6.29
28	*Isolet*	5	-0.004	-0.07	0.066
		10	-0.0008	-0.04	0.0392
29	*Landsat satellite*	5	-0.02	-1.36	1.34
		10	-0.82	-2.41	1.59
30	*Letter recognition*	5	0.0005	-0.03	0.0305
		10	-0.16	-0.12	0.04
31	*Optical digits*	5	-0.006	-0.01	0.016
		10	-0.44	-0.002	0.438
32	*Shuttle*	5	0	0.06	0.06
		10	0.57	0.02	0.55
33	*Person activits*	5	-0.18	0	0.18
		10	0.04	-0.20	0.24
34	*Musk clean*	5	0	-0.38	0.38
		10	-0.0006	-0.18	0.180
35	*Corel image features*	5	0.03	0.21	0.18
		10	0.007	0.03	0.023
36	*The bag of words*	5	-5.20	-0.02	5.18
		10	0.11	-0.09	0.2
37	*Covertype*	5	0.0003	0.09	0.0897
		10	-0.90	-3.38	2.48

Como se muestra en la Tabla 6, se observa que es conveniente usar la variante *Fahim* para resolver instancias grandes cuando se busca buena calidad de agrupación. La variante *Fahim*, obtuvo una diferencia de porcentaje respecto a la variante *O-K-Means* de hasta 12.519% en el mejor de los casos. La variante *O-K-Means* obtuvo dominio frente a la variante *Fahim* en pocos casos, en el mejor de ellos alcanzó un dominio de hasta 5.18%.

Capítulo 5

Conclusiones y trabajos futuros

En este capítulo, se presentan las conclusiones derivadas de los resultados obtenidos con la experimentación realizada durante el presente proyecto de investigación. Asimismo, se exponen algunas propuestas con el fin de continuar ampliando la investigación en este campo con posibles investigaciones posteriores.

5.1 Conclusiones

Al resolver las 37 instancias seleccionadas, con el algoritmo *K-Means* y dos variantes de este algoritmo, se observó que a medida que se incrementaba el valor de $n*d$ también se incrementaba el tiempo de procesamiento. La hipótesis inicial se cumplió al observar que las variantes aplicadas mejoraron notablemente los tiempos de solución y la calidad de agrupación. Las principales contribuciones de esta investigación son las siguientes:

> La primera conclusión, se deriva de su aplicación con respecto a instancias pequeñas. En principio, se determinó $n*d$ = 100,000 como el límite para clasificar las instancias pequeñas. Al agruparlas, la variante *Fahim* destacó porque la agrupación se procesó en menor tiempo al obtener un porcentaje de reducción de hasta 85.32% en el mejor de los casos.

La segunda conclusión, es que, también con instancias pequeñas, las variantes *Fahim* y *O-K-Means* obtuvieron pérdidas de calidad, en comparación con el algoritmo *K-Means* estándar. Al comparar los resultados entre ambas variantes, se identificó menor pérdida de calidad aplicando la variante *O-K-Means* con un porcentaje más bajo de -0.52%. El mejor porcentaje obtenido fue de 1.86%.

La tercera conclusión, está relacionada con grandes instancias. Éstas se clasificaron a partir del valor $n*d = 100{,}000$ porque se observó que su agrupación genera mayor tiempo de procesamiento aplicando el algoritmo *K-Means*. De las variantes aplicadas, se observó que *O-K-Means* fue dominante porque obtuvo menor tiempo de procesamiento, con un porcentaje de reducción de hasta 93.57%.

La cuarta conclusión, se refiere a que la variante *Fahim* es más eficiente porque obtuvo menor pérdida de calidad con instancias grandes. Al comparar los resultados obtenidos entre ambas variantes, se observó que, al aplicar *Fahim*, se obtuvo un porcentaje de pérdida más bajo de -0.903%. Éste resultado mejoró en calidad de agrupación, con 0.71% en el mejor de los casos.

La quinta conclusión, se refiere a las observaciones generales sobre las instancias pequeñas. Se observó un balance entre las variantes con las instancias pequeñas, donde *Fahim* es dominante en el tiempo con 78% del total de instancias pequeñas, y *O-K-Means* es dominante en obtener menor pérdida de calidad con 80% del total de instancias pequeñas.

La sexta conclusión, se refiere a las observaciones sobre los resultados obtenidos con las instancias grandes. Se observó un balance entre las variantes con las instancias grandes, similar al que se obtuvo con las instancias pequeñas. La variante *O-K-Means* fue dominante en obtener menor tiempo de procesamiento con el 75% del total de instancias grandes, y la variante *Fahim* fue dominante en obtener menor pérdida de calidad con 68% del total de instancias grandes.

La séptima conclusión, se refiere a las observaciones realizadas sobre el porcentaje de dominio de la variante *O-K-Means* en el tiempo con respecto a la variante *Fahim*. Se observó que en el mejor de los casos *O-K-Means* resolvió la instancia *Bag of words* con un porcentaje de 144.23% mayor que *Fahim*.

La octava conclusión, se refiere a las observaciones realizadas sobre el porcentaje de dominio de la variante *Fahim* en la calidad de solución con respecto a la variante *O-K-Means*. Se observó que en el mejor de los casos *Fahim* mejoró la calidad de agrupación de la instancia *Magic gamma telescope* con un porcentaje de 12.51% mayor que *O-K-Means.*

Con las mejoras obtenidas al aplicar las variantes *Fahim* y *O-K-Means* del algoritmo *K-Means*, se demostró, que cuando se pretende analizar grandes instancias, la variante *Fahim* ofrece mejores resultados porque la pérdida de calidad es menor. Por otra parte, con la aplicación de la variante *O-K-Means* es factible lograr una mejora del tiempo de ejecución.

5.2 Trabajo futuro

A partir de los resultados obtenidos con la presente investigación, se proponen los siguientes temas:

a) Implementar el principio de la variante K++ en las variantes *Fahim* y *O-K-Means* para tener una mejor inicialización de centroides al solucionar instancias.
b) De acuerdo con la literatura obtenida al elaborar el estado del arte, se identificó que existe un mayor número de mejoras al algoritmo *K-Means* durante la etapa de clasificación. Por esta razón, se propone la aplicación de diferentes variantes del algoritmo *K-Means* para la etapa de clasificación, enfocadas en la solución de grandes instancias.

Referencias

[1] J. Pérez, et al, "Una heurística eficiente aplicada al algoritmo K-Means para el agrupamiento de grandes instancias altamente agrupadas," *Computación y Sistemas ("CyS")*, vol. 22, no 2, pp. 607–619, Feb. 2018.

[2] J. Pérez, et al, "Mejora del algoritmo K-Means mediante una meta-heurística orientada a la reducción de su complejidad computacional," presented at the Encuentro Nacional de Computación (ENC) del Taller sobre Aspectos Algorítmicos de Aplicaciones y Sistemas Computacionales (AAASC), Ocotlán, Oaxaca, México, Nov. 3-5, 2014.

[3] J. Pérez, et al, "An Improvement to the K-Means Algorithm Oriented to Big Data," presented at the International Conference On Numerical Analysis and Applied Mathematics (ICNAAM 2014), Rodas, Grecia, Sep. 22-28, 2014.

[4] J. Pérez, et al, "Balancing effort and benefit of K-Means clustering algorithms in Big Data realms," *PLOS ONE*, vol. 13, no 9, pp. 1-19, Sep. 2018.

[5] J. Pérez, et al, "Optimization of the K-Means algorithm for the solution of high dimensional instances," presented at the International Conference On Numerical Analisys and Applied Mathematics (ICNAAM 2015), Rodas, Grecia, Sep. 23-29, 2016.

[6] R. Basave, "Mejoramiento de la eficiencia y eficacia del algoritmo de agrupamiento K-Means mediante una nueva condición de convergencia," Tesis de maestría, Dept. Ciencias Computacionales, CENIDET, Cuernavaca, Morelos, 2005.

[7] J. Moreno, "Estudio e implementación de las mejoras más relevantes del algoritmo K-Means y su análisis comparativo," Tesis de maestría, Dept. Ciencias Computacionales, CENIDET, Cuernavaca, Morelos, 2016.

[8] N. Almanza, "Desarrollo de heurísticas para la mejora del algoritmo K-Means en las fases de clasificación y convergencia," Tesis de doctorado, Dept. Ciencias Computacionales, CENIDET, Cuernavaca, Morelos, 2018.

[9] A. Díaz. "Desarrollo de una mejora al algoritmo K-Means orientadas al paradigma de Big Data," Tesis de maestría, Dept. Ciencias Computacionales, CENIDET, Cuernavaca, Morelos, 2018.

[10] K. Patel y P. Thakral, "The best clustering algorithms in data mining," presented at the International Conference on Communication and Signal Processing (ICCSP), Tamil Nadu, India, Abr. 6-8, 2016.

[11] J. Pérez, et al, "Mejora al algoritmo de agrupamiento K-Means mediante un nuevo criterio de convergencia y su aplicación a bases de datos poblacionales de cáncer," presented at the 2do Taller Latino Iberoamericano de Investigación de Operaciones, Acapulco, Guerrero, México, Oct. 3-5, 2007.

[12] S. Shah y M. Singh, "Comparison of a time efficient modified k-mean algorithm with K-Mean and K-Medoid algorithm," presented at the International Conference on

Communication Systems and Network Technologies, Rajkot, Gujrat, India, May. 11-13, 2012.

[13] D. Napoleon y P. Ganga, "An efficient K-Means clustering algorithm for reducing time complexity using uniform distribution data points," presented at the Trendz in Information Sciences & Computing(TISC2010), Chennai, India, Dic. 17-19, 2010.

[14] A. Mohammed, A. Wesam, "Efficient and Fast Initialization Algorithm for Kmeans Clustering," *International Journal of Intelligent Systems and Applications (IJISA)*, vol. 4, no. 1, pp 21-31, 2012.

[15] L. Zhang, et al, "Improvement of K-Means algorithm based on density," presented at the IEEE 8th Joint International Information Technology and Artificial Intelligence Conference, Chongqing, China, May. 24-26, 2019.

[16] M. Chiang, et al, "A Time-Efficient pattern reduction algorithm for K-Means clustering," *Information Sciences*, vol. 181, no. 4, pp. 716-731, 2011.

[17] J. Pérez, et al, "A-means: improving the cluster assignment phase of K-Means for Big Data," *International Journal of Combinatorial Optimization Problems and Informatics*, vol. 9, no 2, pp. 3-10, Feb. 2018.

[18] J. Pérez et al, "Early classification: A new heuristic to improve the classification step of K-Means," *Journal of Information and Data Management*, vol. 4, no. 2, pp. 94-103, 2013.

[19] A. Fahim, et al, "An efficient enhanced K-Means clustering algorithm," *Journal of Zhejiang University-SCIENCE A*, vol. 7 no. 10, pp. 1626-1633, 2006.

[20] L. Kaufman and P. Rousseeuw, "CLARA in R: Clustering large application" in *Finding Groups in Data: An Introduction to Cluster Analysis*, Wiley 1990, New York, USA, 1990, pp 126-163.

[21] A. Mexicano, et al, "The early stop heuristic: A new convergence criterion for K-Means," presented at the International Conference of Numerical Analysis and Applied Mathematics, Rodas, Grecia, Sep. 23-29, 2015.

[22] D. Arthur and S. Vassilvitskii, "k-means++: The adventages of careful seeding," presented at the Eighteenth anual ACM-SIAM symposium on discrete algorithms, Nueva Orleans, Luisiana, USA, Ene. 7-9, 2007.

[23] A. Hernández. "Optimización del algoritmo K-Means orientada a Big Data mediante la integración de heurísticas en las fases de clasificación y convergencia," Tesis de maestría, Dept. Ciencias Computacionales, CENIDET, Cuernavaca, Morelos, 2017.

[24] C. Xiaoli, et al, "Optimized big data K-Means clustering using MapReduce," *The Journal of Supercomputing*, vol. 70, no. 3, pp 1249-1259, 2014.

[25] B. Bahmani, et al, "Scalable K-Means++," *Proceedings of the VLDB Endowment (PVLDB)*, vol. 5, no. 7, pp. 622-633, 2012.

[26] D. Graff, et al, “UCI-Machine learning repository,” https://archive.ics.uci.edu/ml/datasets.php. (accessed Enero 1, 2020).

[27] J. Alcalá, et al, “KEEL-dataset,” https://sci2s.ugr.es/keel/datasets.php (accessed Enero 1, 2020).

[28] X. Lan, et al, “Density K-Means: A new algorithm for centers initialization for K-Means,” presented at the IEEE International Conference on Software Engineering and Service Science (ICSESS), Beijing, China, Sep. 23-25, 2015.

[29] B. Kulis and M. Jordan, “Revisting K-Means: New algorithms via bayesian nonparametrics,” presented at the 29 th International Conference on Machine Learning, Edimburgo, Escocia, Reino Unido, 26 Jun. – 1 Jul. 2012.

[30] P. Rathore and D. Shukla, “Analysis and performance improvement of K-Means in big data environment,” presented at the international Conference on Communication Networks (ICCN), Gwalior, Madhya Pradesh, India, Nov. 19-21, 2015.

[31] S. Chakraborty and S. Das. “K-Means clustering with a new divergence based distance metric: Convergence and performance analysis,” *Pattern Recognition Letters* vol. 100, no. C, pp. 67-63, 2017.

[32] M. Emre, et al, “A comparative study of efficient initialization methods for the K-Means clustering algorithm,” *Expert Systems with Applications*, vol. 40, no. 1, pp. 200-210, 2013.

[33] S. Lloyd, “Leats squares quantization in PCM,” *Transactions on Information Therory*, vol. 28, no. 2, pp. 129-137, 1982.

Anexos

Anexo A: Resultados experimentales con variante *Fahim*

En la Tabla A.1, se muestran los tiempos que le tomó a la variante *Fahim* al resolver cada instancia en el experimento A. La primera columna contiene el identificador para cada instancia, la segunda columna contiene el nombre de la instancia que se resolvió, la tercera columna contiene el tiempo en milisegundos (T (ms)) que le tomó al algoritmo *K-Means* estándar para resolver la instancia dada, la cuarta columna contiene la cantidad de iteraciones (I) que realizó el algoritmo *K-Means* estándar, la quinta columna contiene el tiempo en milisegundos que le tomó a la variante *Fahim* (Tf (ms)) para resolver la instancia dada, la sexta columna contiene la cantidad de iteraciones (If) que realizó la variante *Fahim*, la séptima columna contiene el porcentaje de diferencia en el tiempo (Tf (%)) entre el algoritmo *K-Means* estándar y la variante *Fahim*. Es conveniente mencionar que un valor positivo en la última columna, significa que la variante *Fahim* resolvió la instancia dada en menor tiempo que el algoritmo *K-Means* estándar. Por el contrario, un valor negativo significa que la variante *Fahim* resolvió la instancia dada en mayor tiempo que el algoritmo *K-Means* estándar.

Tabla A.1. Ventaja de *Fahim* sobre tiempo en el experimento A.

ID	Instancia	T (ms)	I	Tf (ms)	If	Tf (%)
1	*Iris*	0.6990	4	0.3218	5	53.95
2	*Appendicitis*	1.3580	7	0.6161	7	54.63
3	*Newthyroid*	2.4991	14	1.1051	8	55.78
4	*Glass identification*	1.8117	6	1.9460	8	-7.04
5	*Bupa*	6.5181	17	5.1460	21	21.05
6	*Wine*	5.2700	8	1.2540	15	76.20
7	*Ecoli*	2.4619	7	2.2280	12	9.50
8	*Balance scale*	6.7391	16	7.2629	26	-7.77
9	*Heart disease*	14.3750	23	9.3880	23	34.69
10	*Cleveland*	9.8860	15	4.9889	11	49.53
11	*Steel plates faults*	230.0889	28	169.7950	29	26.20
12	*Pima*	14.8501	14	8.1069	11	45.40
13	*Pima indians*	9.4891	9	7.6730	11	19.13
14	*Breast cancer wisconsin*	27.8411	27	18.9600	30	31.89
15	*Concrete data*	42.8181	17	10.9971	10	74.31
16	*Cloud*	77.3799	47	43.4110	39	43.89
17	*Spectf*	29.5508	16	26.6621	22	9.77
18	*Yeast*	31.3320	16	32.7780	23	-4.61

19	*Vehicle silhouettes*	54.0440	22	32.1500	23	40.51
20	*Abalone*	208.8208	44	147.1360	44	29.53
21	*Image segmentation*	204.9699	31	79.8969	18	61.02
22	*Page blocks clasification*	330.6341	38	261.1041	38	21.02
23	*Wine quality*	631.0010	48	92.6061	11	85.32
24	*Parkinsons*	1,041.4271	56	405.6199	33	61.05
25	*Wall following*	550.4060	24	406.2970	26	26.18
26	*Pen digits*	505.9650	17	331.3069	16	34.51
27	*Magic gamma telescope*	978.2481	30	821.4102	35	16.03
28	*Isolet*	1,396.9299	39	1,100	45	21.20
29	*Landsat satellite*	779.7549	22	616.4920	26	20.93
30	*Letter recognition*	1,854.3291	33	1,377.5651	35	25.71
31	*Optical digits*	1,576.4310	28	1,237.5080	32	21.49
32	*Shuttle*	2,113.7850	24	1,567.8391	24	25.82
33	*Person Activitis*	626.4501	6	178.2141	2	71.55
34	*Musk clean*	2,364.4161	13	1,253.4220	14	46.98
35	*Corel image features*	35,217.1507	36	36,749.1540	52	-4.35
36	*Bag of words*	65,253.0050	36	66,194.2601	51	-1.44
37	*Covertype*	462.441.2479	85	280,185.5859	88	39.41

En la Tabla A.2, se muestran los tiempos que le tomó a la variante *Fahim* al resolver cada instancia en el experimento B. La primera columna contiene el identificador para cada instancia, la segunda columna contiene el nombre de la instancia que se resolvió, la tercera columna contiene el tiempo en milisegundos que le tomó al algoritmo *K-Means* estándar para resolver la instancia dada, la cuarta columna contiene la cantidad de iteraciones que realizó el algoritmo *K-Means* estándar, la quinta columna contiene el tiempo en milisegundos que le tomó a la variante *Fahim* para resolver la instancia dada, la sexta columna contiene la cantidad de iteraciones que realizó la variante *Fahim*, la séptima columna contiene el porcentaje de diferencia en el tiempo entre el algoritmo *K-Means* estándar y la variante *Fahim*. Es conveniente mencionar que un valor positivo en la última columna, significa que la variante *Fahim* resolvió la instancia dada en menor tiempo que el algoritmo *K-Means* estándar. Por el contrario, un valor negativo significa que la variante *Fahim* resolvió la instancia dada en mayor tiempo que el algoritmo *K-Means* estándar.

Tabla A.2. Ventaja de *Fahim* sobre tiempo en el experimento B.

Id	Instancia	T (ms)	I	Tf (ms)	If	Tf (%)
1	*Iris*	2.3568	12	0.6580	7	72.07
2	*Appendicitis*	1.6780	7	0.6471	5	61.43
3	*Newthyroid*	6.1500	17	2.9740	15	51.64
4	*Glass identification*	6.6979	11	4.5340	11	32.30
5	*Bupa*	9.8262	14	5.1792	14	47.29
6	*Wine*	15.0530	20	5.6520	16	62.45
7	*Ecoli*	14.5700	21	5.5449	15	61.94
8	*Balance scale*	10.8471	13	5.1780	11	52.26
9	*Heart disease*	38.6639	32	11.9998	20	68.96
10	*Cleveland*	22.0971	17	8.9970	13	59.28
11	*Steel plates faults*	1,075.2248	67	377.2762	40	64.91
12	*Pima*	45.4030	22	28.0888	24	38.13
13	*Pima indians*	33.1280	16	17.8909	17	45.99
14	*Breast cancer wisconsin*	58.2490	29	16.3181	15	71.98
15	*Concrete data*	91.7299	22	21.3161	14	76.76
16	*Cloud*	0.1426	43	69.8559	40	51.00
17	*Spectf*	44.2519	12	30.8330	17	30.32
18	*Yeast*	178.7360	48	76.8659	36	56.99
19	*Vehicle silhouettes*	78.0730	16	39.6459	16	49.21
20	*Abalone*	894.7530	94	186.5640	40	79.14
21	*Image segmentation*	420.2361	32	196.240	29	53.30
22	*Page blocks clasification*	1,616.6852	94	578.2409	54	64.23
23	*Wine quality*	1,226.0289	48	647.0850	46	47.22
24	*Parkinsons*	1,642.8561	45	731.4839	41	55.47
25	*Wall following*	1,226.7329	27	889.4360	45	27.49
26	*Pen digits*	3,164.7861	54	1,337.0821	46	57.75
27	*Magic gamma telescope*	7,453.9771	117	2,775.7101	75	62.76
28	*Isolet*	3,369.1849	48	1,954.5228	48	41.98
29	*Landsat satellite*	4,890.0489	70	2,181.0338	63	55.39

30	*Letter recognition*	6,754.0090	62	3,230.6800	49	52.16
31	*Optical digits*	2,669.0787	25	1,209.7611	23	55.17
32	*Shuttle*	7,838.0579	45	4,216.6719	40	46.20
33	*Person Activitis*	6,801.8488	33	1,785.1040	33	73.75
34	*Musk clean*	5,011.7369	14	2,364.4600	15	52.82
35	*Corel image features*	206,799.7989	110	121,407.0780	109	41.29
36	*Bag of words*	291,284.7528	81	466,800.0450	204	-60.25
37	*Covertype*	3,194,870.1789	209	524,623.9340	98	83.57

En la Tabla A.3, se muestran resultados de la función objetivo que la variante *Fahim* obtuvo al resolver cada instancia en el experimento A. La primera columna contiene el identificador para cada instancia, la segunda columna contiene el nombre de la instancia que se resolvió, la tercera columna contiene el valor de la función objetivo (Z) que algoritmo *K-Means* estándar obtuvo al resolver la instancia dada, la cuarta columna contiene la cantidad de iteraciones que realizó el algoritmo *K-Means* estándar, la quinta columna contiene la función objetivo (Zf) que la variante *Fahim* obtuvo al resolver la instancia dada, la sexta columna contiene la cantidad de iteraciones que realizó la variante *Fahim*, la séptima columna contiene el porcentaje de diferencia en la calidad de agrupación (Zf (%)) entre el algoritmo *K-Means* estándar y la variante *Fahim*. Es conveniente mencionar que un valor positivo en la última columna, significa que la variante *Fahim* resolvió la instancia dada con un valor en la función objetivo menor al que obtuvo el algoritmo *K-Means* estándar. Por el contrario, un valor negativo significa que la variante *Fahim*, resolvió la instancia dada con un valor en la función objetivo mayor al que obtuvo el algoritmo *K-Means* estándar.

Tabla A.3. Ventaja de *Fahim* sobre calidad en el experimento A.

ID	Instancia	Z	I	Zf	If	Zf (%)
1	*Iris*	76.6213	4	80.7098	5	-5.33
2	*Appendicitis*	27.6226	7	27.6226	7	0
3	*Newthyroid*	1,475.1251	14	1,616.3263	8	-9.57
4	*Glass identification*	231.2173	6	230.4377	8	0.33
5	*Bupa*	7,399.0044	17	7,405.1210	21	-0.08
6	*Wine*	13,142.6537	8	17,160.1758	15	-30.56
7	*Ecoli*	73.2410	7	85.9075	12	-17.29
8	*Balance scale*	1,244.8007	16	1,245.9275	26	-0.09
9	*Heart disease*	10,722.3289	23	10,736.3833	23	-0.13
10	*Cleveland*	9,148.2341	15	9,122.3696	11	0.28

11	*Steel plates faults*	1,140,603,847.4474	28	1,140,260,193.6917	29	0.03
12	*Pima*	32,554.0621	14	32,846.6394	11	-0.89
13	*Pima indians*	32,724.6080	9	33,161.5956	11	-1.33
14	*Breast cancer wisconsin*	2,513.4281	27	2,531.5954	30	-0.72
15	*Concrete data*	135,743.0347	17	135,800.3958	10	-0.04
16	*Cloud*	109,659.1908	47	107,665.4404	39	1.81
17	*Spectf*	11,998.7554	16	12,156.558	22	-1.31
18	*Yeast*	271.4353	16	271.3799	23	0.02
19	*Vehicle silhouettes*	40.732.3483	22	42,779.7849	23	-5.02
20	*Abalone*	621.0059	44	621.0059	44	0
21	*Image segmentation*	170,848.2657	31	171,550.0888	18	-0.41
22	*Page blocks clasification*	3,530,905.8006	38	3,497,151.5879	38	0.95
23	*Wine quality*	107,863.5412	48	116,518.1503	11	-8.02
24	*Parkinsons*	111,553.7438	56	111,867.3989	33	-0.28
25	*Wall following*	27,034.0561	24	27,033.5583	26	0.001
26	*Pen digits*	874,999.2295	17	874,997.7668	16	0.0001
27	*Magic gamma telescope*	1,262,793.5035	30	1,262,811.2413	35	-0.001
28	*Isolet*	11,644.1100	39	11,644.5871	45	-0.004
29	*Landsat satellite*	311,537.5065	22	311,605.1237	26	-0.02
30	*Letter recognition*	142,154.4750	33	142,155.2632	35	0.0005
31	*Optical digits*	159,468.9816	28	159,479.7090	32	-0.006
32	*Shuttle*	1,727,377.5050	24	1,727,377.5050	24	0
33	*Person Activitis*	1,500,035,006,970,350	6	1,502,820,359,956,090	2	-0.18
34	*Musk clean*	4,913,622.9455	13	4,913,622.9455	14	0
35	*Corel image features*	248,675.4585	36	248,582.0999	52	0.03
36	*Bag of words*	20,231,766,994.1834	36	20,231,768,047.0755	51	-5.20
37	*Covertype*	1,603,806,112.9100	85	1,603,801,109.8632	88	0.0003

En la Tabla A.4, se muestran resultados de la función objetivo que la variante *Fahim* obtuvo al resolver cada instancia en el experimento B. La primera columna contiene el identificador para cada instancia, la segunda columna contiene el nombre de la instancia que se resolvió, la tercera columna contiene el valor de la función objetivo que algoritmo *K-Means* estándar obtuvo al resolver la instancia dada, la cuarta columna contiene la cantidad de iteraciones que realizó el algoritmo *K-Means* estándar, la quinta columna contiene la función objetivo que la variante *Fahim* obtuvo al resolver la instancia dada, la sexta columna contiene la cantidad de iteraciones que realizó la variante *Fahim*, la séptima columna contiene el porcentaje de diferencia en la calidad de agrupación entre el algoritmo *K-Means* estándar y la variante *Fahim*. Es conveniente mencionar que un valor positivo en la última columna, significa que la variante *Fahim* resolvió la instancia dada con un valor en la función objetivo menor al que obtuvo el algoritmo *K-Means* estándar. Por el contrario, un valor negativo significa que la variante *Fahim*, resolvió la instancia dada con un valor en la función objetivo mayor al que obtuvo el algoritmo *K-Means* estándar.

Tabla A.4. Ventaja de *Fahim* sobre calidad en el experimento B.

ID	Instancia	Z	I	Zf	If	Zf (%)
1	*Iris*	59.3594	12	61.5991	7	-3.77
2	*Appendicitis*	21.0976	7	21.3846	5	-1.36
3	*Newthyroid*	1,092.5131	17	1,113.6642	15	-1.93
4	*Glass identification*	184.4912	11	193.0970	11	-4.66
5	*Bupa*	5,919.5346	14	6,193.7912	14	-4.63
6	*Wine*	5,947.7059	20	6,502.9717	16	-9.33
7	*Ecoli*	61.6012	21	61.6144	15	-0.02
8	*Balance scale*	1,004.8556	13	1,006.3839	11	-0.15
9	*Heart disease*	8.488.6902	32	8,730.6711	20	-2.85
10	*Cleveland*	7,301.3564	17	7,447.2147	13	-1.99
11	*Steel plates faults*	720,553,777.3929	67	748,758,341.5924	40	-3.91
12	*Pima*	24,538.7353	22	24,524.6908	24	0.05
13	*Pima indians*	24,295.2153	16	24,470.2677	17	-0.72
14	*Breast cancer wisconsin*	2,342.4989	29	2,373.4101%	15	-1.31
15	*Concrete data*	105,863.5566	22	107,049.5228	14	-1.12
16	*Cloud*	64,719.6181	43	65,054.4978	40	-0.51
17	*Spectf*	11,140.7705	12	11,138.9370	17	0.01
18	*Yeast*	241.4025	48	240.162	36	0.51

19	*Vehicle silhouettes*	30,342.2189	16	30,474.6103	16	-0.43
20	*Abalone*	410.4356	94	414.0875	40	-0.88
21	*Image segmentation*	128,623.4992	32	130,528.5513	29	-1.48
22	*Page blocks clasification*	2,017,305.5048	94	2,080,648.7522	54	-3.13
23	*Wine quality*	80,313.2897	48	80,307.3072	46	0.007
24	*Parkinsons*	90,310.6677	45	90,761.6570	41	-0.49
25	*Wall following*	24,785.3782	27	24,792.1348	45	-0.02
26	*Pen digits*	693,024.1306	54	693,022.3472	46	0.0002
27	*Magic gamma telescope*	1,107,669.4916	117	1,099,715.4627	75	0.71
28	*Isolet*	10,159.7752	48	10,159.8585	48	-0.0008
29	*Landsat satellite*	254,754.3169	70	256,848.0152	63	-0.82
30	*Letter recognition*	126,739.2373	62	126,950.3294	49	-0.16
31	*Optical digits*	140,576.4556	25	141,205.6161	23	-0.44
32	*Shuttle*	1,419,331.0011	45	1,411,154.3577	40	0.57
33	*Person Activitis*	586,257,687,229,319	33	585,985,748,044,461	33	0.04
34	*Musk clean*	4,634,558.3745	14	4,634,588.3265	15	-0.0006
35	*Corel image features*	222,677.0936	110	222,660.6194	109	0.007
36	*Bag of words*	14,466,965,377.3437	81	14,450,103,709.1473	204	0.11
37	*Covertype*	1,405,197,167.1200	209	1,417,890,000.0557	98	-0.90

Anexo B: Resultados experimentales con variante O-K-Means

En la Tabla B.1, se muestran los tiempos que le tomó a la variante *O-K-Means* al resolver cada instancia en el experimento A. La primera columna contiene el identificador para cada instancia, la segunda columna contiene el nombre de la instancia que se resolvió, la tercera columna contiene el tiempo en milisegundos que le tomó al algoritmo *K-Means* estándar para resolver la instancia dada, la cuarta columna contiene la cantidad de iteraciones que realizó el algoritmo *K-Means* estándar, la quinta columna contiene el tiempo en milisegundos (To (ms)) que le tomó a la variante *O-K-Means* para resolver la instancia dada, la sexta columna contiene la cantidad de iteraciones (Io) que realizó la variante *O-K-Means*, la séptima columna contiene el porcentaje de diferencia (To (%)) en el tiempo entre el algoritmo *K-Means* estándar y la variante *O-K-Means*. Es conveniente mencionar que un valor positivo en la última columna, significa que la variante *O-K-Means* resolvió la instancia dada en menor tiempo que el algoritmo *K-Means* estándar. Por el contrario, un valor negativo significa que la variante *O-K-Means* resolvió la instancia dada en mayor tiempo que el algoritmo *K-Means* estándar.

Tabla B.1. Ventaja de *O-K-Means* sobre tiempo en el experimento A.

ID	Instancia	T (ms)	I	To (ms)	Io	To (%)
1	*Iris*	0.6990	4	0.4089	4	41.50
2	*Appendicitis*	1.3880	7	0.8499	7	-2.59
3	*Newthyroid*	2.4991	14	2.34	13	6.01
4	*Glass identification*	1.8117	6	1.8100	6	0.43
5	*Bupa*	6.5181	17	6.1659	17	5.40
6	*Wine*	5.2700	8	5.5150	14	-4.65
7	*Ecoli*	2.4619	7	5.5758	6	61.73
8	*Balance scale*	6.7391	16	5.9008	14	12.43
9	*Heart disease*	14.3750	23	12.3910	20	13.80
10	*Cleveland*	9.8860	15	7.8368	12	20.72
11	*Steel plates faults*	230.0889	28	172.2030	21	25.15
12	*Pima*	14.8501	14	10.6739	10	28.12
13	*Pima indians*	9.4891	9	5.2909	5	44.24
14	*Breast cancer wisconsin*	27.8411	27	14.5258	14	47.82
15	*Concrete data*	42.8181	17	14.7128	10	65.63
16	*Cloud*	77.3799	47	40.6558	25	47.49
17	*Spectf*	29.5508	16	29.9661	16	-1.40

18	*Yeast*	31.3320	16	19.6769	10	37.19
19	*Vehicle silhouettes*	54.0440	22	37.1520	15	31.25
20	*Abalone*	208.8208	44	136.9040	29	34.43
21	*Image segmentation*	204.9699	31	73.7779	11	64.00
22	*Page blocks clasification*	330.6341	38	202.6028	23	38.72
23	*Wine quality*	631.0010	48	329.2620	25	47.81
24	*Parkinsons*	1,041.4271	56	298.1450	16	71.37
25	*Wall following*	550.4060	24	344.4390	15	37.42
26	*Pen digits*	505.9650	17	186.4829	6	63.14
27	*Magic gamma telescope*	978.2481	30	893.9149	3	8.62
28	*Isolet*	1,396.9299	39	434.2272	12	68.91
29	*Landsat satellite*	779.7549	22	295.7749	8	62.06
30	*Letter recognition*	1,854.3291	33	854.1688	15	53.93
31	*Optical digits*	1,576.4310	28	1,173.2659	20	25.57
32	*Shuttle*	2,113.7850	24	1,338.5251	15	36.67
33	*Person Activitis*	626.4501	6	630.2759	6	-0.61
34	*Musk clean*	2,364.4161	13	912.6510	5	61.40
35	*Corel image features*	35,217.1507	36	10,833.5428	11	69.23
36	*Bag of words*	65,253.0050	36	16,327.5198	9	74.97
37	*Covertype*	462.441.2479	85	45,642.8678	9	90.13

En la Tabla B.2, se muestran los tiempos que le tomó a la variante *O-K-Means* al resolver cada instancia en el experimento B. La primera columna contiene el identificador para cada instancia, la segunda columna contiene el nombre de la instancia que se resolvió, la tercera columna contiene el tiempo en milisegundos que le tomó al algoritmo *K-Means* estándar para resolver la instancia dada, la cuarta columna contiene la cantidad de iteraciones que realizó el algoritmo *K-Means* estándar, la quinta columna contiene el tiempo en milisegundos que le tomó a la variante *O-K-Means* para resolver la instancia dada, la sexta columna contiene la cantidad de iteraciones que realizó la variante *O-K-Means*, la séptima columna contiene el porcentaje de diferencia en el tiempo entre el algoritmo *K-Means* estándar y la variante *O-K-Means*. Es conveniente mencionar que un valor positivo en la última columna, significa que la variante *O-K-Means* resolvió la instancia dada en menor tiempo que el algoritmo *K-Means*

estándar. Por el contrario, un valor negativo significa que la variante *O-K-Means* resolvió la instancia dada en mayor tiempo que el algoritmo *K-Means* estándar.

Tabla B.2. Ventaja de *O-K-Means* sobre tiempo en el experimento B.

ID	Instancia	T	I	To	Io	To (%)
1	*Iris*	2.3568	12	2.1670	11	8.05
2	*Appendicitis*	1.6780	7	1.6758	7	-1.36
3	*Newthyroid*	6.1500	17	5.4941	15	10.66
4	*Glass identification*	6.6979	11	6.6738	11	0.35
5	*Bupa*	9.8262	14	5,929.0037	7	49.25
6	*Wine*	15.0530	20	10.6082	14	29.52
7	*Ecoli*	14.5700	21	5.5759	8	61.73
8	*Balance scale*	10.8471	13	10.9279	13	-0.74
9	*Heart disease*	38.6639	32	30.9479	26	19.95
10	*Cleveland*	22.0971	17	19.4550	15	11.95
11	*Steel plates faults*	1,075.2248	67	866.1470	54	19.44
12	*Pima*	45.4030	22	39.6061	19	12.76
13	*Pima indians*	33.1280	16	24.8749	12	24.91
14	*Breast cancer wisconsin*	58.2490	29	50.3681	25	13.52
15	*Concrete data*	91.7299	22	52.7000	18	42.54
16	*Cloud*	0.1426	43	121.7580	38	14.59
17	*Spectf*	44.2519	12	44.4920	12	-0.54
18	*Yeast*	178.7360	48	47.5950	13	73.37
19	*Vehicle silhouettes*	78.0730	16	44.3292	9	43.22
20	*Abalone*	894.7530	94	267.8730	29	70.06
21	*Image segmentation*	420.2361	32	275.2039	21	34.51
22	*Page blocks clasification*	1,616.6852	94	744.7011	43	53.93
23	*Wine quality*	1,226.0289	48	669.5259	26	45.39
24	*Parkinsons*	1,642.8561	45	986.7511	27	39.93
25	*Wall following*	1,226.7329	27	453.0790	10	63.06
26	*Pen digits*	3,164.7861	54	601.1970	10	81.00
27	*Magic gamma telescope*	7,453.9771	117	4,151.0000	7	44.31
28	*Isolet*	3,369.1849	48	1,623.8079	23	51.80

29	*Landsat satellite*	4,890.0489	70	1,254.8401	17	74.33
30	*Letter recognition*	6,754.0090	62	4,598.6741	42	31.91
31	*Optical digits*	2,669.0787	25	2,256.5329	20	16.39
32	*Shuttle*	7,838.0579	45	3,868.5131	22	50.64
33	*Person Activitis*	6,801.8488	33	1,445.9199	7	78.74
34	*Musk clean*	5,011.7369	14	3,222.1820	9	35.70
35	*Corel image features*	206,799.7989	110	62,446.4540	33	69.80
36	*Bag of words*	291,284.7528	81	46,642.7631	13	83.98
37	*Covertype*	3,194,870.1789	209	205,370.0249	21	93.57

En la Tabla B.3, se muestran resultados de la función objetivo que la variante *O-K-Means* obtuvo al resolver cada instancia en el experimento A. La primera columna contiene el identificador para cada instancia, la segunda columna contiene el nombre de la instancia que se resolvió, la tercera columna contiene el valor de la función objetivo que algoritmo *K-Means* estándar obtuvo al resolver la instancia dada, la cuarta columna contiene la cantidad de iteraciones que realizó el algoritmo *K-Means* estándar, la quinta columna contiene la función objetivo que la variante *O-K-Means* obtuvo al resolver la instancia dada, la sexta columna contiene la cantidad de iteraciones que realizó la variante *O-K-Means*, la séptima columna contiene el porcentaje de diferencia en la calidad de agrupación entre el algoritmo *K-Means* estándar y la variante *O-K-Means*. Es conveniente mencionar que un valor positivo en la última columna, significa que la variante *O-K-Means* resolvió la instancia dada con un valor en la función objetivo menor al que obtuvo el algoritmo *K-Means* estándar. Por el contrario, un valor negativo significa que la variante *O-K-Means*, resolvió la instancia dada con un valor en la función objetivo mayor al que obtuvo el algoritmo *K-Means* estándar.

Tabla B.3. Ventaja de *O-K-Means* sobre calidad en el experimento A.

ID	Instancia	Z	I	Zo	Io	Zo (%)
1	*Iris*	76.6213	4	76.6213	4	0
2	*Appendicitis*	28.4445	7	28.4445	7	0
3	*Newthyroid*	1,475.1251	14	1,475.1251	13	0
4	*Glass identification*	231.2173	6	231.2173	6	0
5	*Bupa*	7,399.0044	17	7,399.0044	17	0
6	*Wine*	13,142.6537	8	13,142.6537	14	0
7	*Ecoli*	73.2410	7	73.2410	6	0
8	*Balance scale*	1,244.8007	16	1,244.9035	14	-0.008
9	*Heart disease*	10,722.3289	23	10,718.4627	20	0.03
10	*Cleveland*	9,148.2341	15	9,154.1134	12	-0.06

11	*Steel plates faults*	1,140,603,847.4474	28	1,137,250,778.6880	21	0.29
12	*Pima*	32,554.0621	14	32,556.5293	10	-0.007
13	*Pima indians*	32,724.6080	9	32,730.0259	5	-0.01
14	*Breast cancer wisconsin*	2,513.4281	27	2,524.5368	14	-0.44
15	*Concrete data*	135,743.0347	17	135,767.7305	10	-0.01
16	*Cloud*	109,659.1908	47	107,616.2011	25	1.86
17	*Spectf*	11,998.7554	16	11,998.7554	16	0
18	*Yeast*	271.4353	16	271.3799	10	0.004
19	*Vehicle silhouettes*	40.732.3483	22	40,818.9086	15	-0.21
20	*Abalone*	621.0059	44	624.9395	29	-0.63
21	*Image segmentation*	170,848.2657	31	170,742.0340	11	0.06
22	*Page blocks clasification*	3,530,905.8006	38	3,418,246.8325	23	3.19
23	*Wine quality*	107,863.5412	48	108,432.9310	25	-0.52
24	*Parkinsons*	111,553.7438	56	114,604.8490	16	-2.73
25	*Wall following*	27,034.0561	24	27,035.1806	15	-0.004
26	*Pen digits*	874,999.2295	17	875,122.8654	6	-0.01
27	*Magic gamma telescope*	1,262,793.5035	30	1,420,935.8644	3	-12.52
28	*Isolet*	11,644.1100	39	11,652.5927	12	-0.07
29	*Landsat satellite*	311,537.5065	22	315,794.1049	8	-1.36
30	*Letter recognition*	142,154.4750	33	142,210.9166	15	-0.03
31	*Optical digits*	159,468.9816	28	159,488.9016	20	-0.01
32	*Shuttle*	1,727,377.5050	24	1,726,178.2473	15	0.06
33	*Person Activitis*	1,500,035,006,970,350	6	1,500,035,006,970,350	6	0
34	*Musk clean*	4,913,622.9455	13	4,932,761.7272	5	-0.38
35	*Corel image features*	248,675.4585	36	248,141.6941	11	0.21
36	*Bag of words*	20,231,766,994.1834	36	20,236,699,936.7078	9	-0.02
37	*Covertype*	1,603,806,112.9100	85	1,602,294,226.9100	9	0.09

En la Tabla B.4, se muestran resultados de la función objetivo que la variante *O-K-Means* obtuvo al resolver cada instancia en el experimento B. La primera columna contiene el identificador para cada instancia, la segunda columna contiene el nombre de la instancia que

se resolvió, la tercera columna contiene el valor de la función objetivo que algoritmo *K-Means* estándar obtuvo al resolver la instancia dada, la cuarta columna contiene la cantidad de iteraciones que realizó el algoritmo *K-Means* estándar, la quinta columna contiene la función objetivo que la variante *O-K-Means* obtuvo al resolver la instancia dada, la sexta columna contiene la cantidad de iteraciones que realizó la variante *O-K-Means*, la séptima columna contiene el porcentaje de diferencia en la calidad de agrupación entre el algoritmo *K-Means* estándar y la variante *O-K-Means*. Es conveniente mencionar que un valor positivo en la última columna, significa que la variante *O-K-Means* resolvió la instancia dada con un valor en la función objetivo menor al que obtuvo el algoritmo *K-Means* estándar. Por el contrario, un valor negativo significa que la variante *O-K-Means*, resolvió la instancia dada con un valor en la función objetivo mayor al que obtuvo el algoritmo *K-Means* estándar.

Tabla B.4. Ventaja de *O-K-Means* sobre calidad en el experimento B.

ID	Instancia	Z	I	Zo	Io	Zo (%)
1	*Iris*	59.3594	12	59.3594	11	0
2	*Appendicitis*	21.0976	7	21.0976	7	0
3	*Newthyroid*	1,092.5131	17	1,092.9018	15	-0.03
4	*Glass identification*	184.4912	11	184.4912	11	0
5	*Bupa*	5,919.5346	14	5,929.0037	7	-0.15
6	*Wine*	5,947.7059	20	6,287.8316	14	-5.71
7	*Ecoli*	61.6012	21	61.7133	8	-0.18
8	*Balance scale*	1,004.8556	13	1,004.8556	13	0
9	*Heart disease*	8.488.6902	32	8,518.8599	26	-0.35
10	*Cleveland*	7,301.3564	17	7,306.5245	15	-0.07
11	*Steel plates faults*	720,553,777.3929	67	718,319,029.3192	54	0.31
12	*Pima*	24,538.7353	22	24,558.7943	19	-0.08
13	*Pima indians*	24,295.2153	16	24,315.6646	12	-0.08
14	*Breast cancer wisconsin*	2,342.4989	29	2,343.7609	25	-0.05
15	*Concrete data*	105,863.5566	22	106,024.6023	18	-0.15
16	*Cloud*	64,719.6181	43	64,723.5297	38	-0.006
17	*Spectf*	11,140.7705	12	11,140.7705	12	0
18	*Yeast*	241.4025	48	244.1457	13	-1.13
19	*Vehicle silhouettes*	30,342.2189	16	30,397.6291	9	-0.18
20	*Abalone*	410.4356	94	415.2089	29	-1.16

21	*Image segmentation*	128,623.4992	32	128,745.7615	21	-0.09
22	*Page blocks clasification*	2,017,305.5048	94	2,077,810.5072	43	-2.99
23	*Wine quality*	80,313.2897	48	80,262.3343	26	0.06
24	*Parkinsons*	90,310.6677	45	90,499.4765	27	-0.20
25	*Wall following*	24,785.3782	27	24,796.0089	10	-0.04
26	*Pen digits*	693,024.1306	54	704,635.7416	10	-1.67
27	*Magic gamma telescope*	1,107,669.4916	117	1,169,495.4116	7	-5.58
28	*Isolet*	10,159.7752	48	10,164.4551	23	-0.04
29	*Landsat satellite*	254,754.3169	70	260,898.3441	17	-2.41
30	*Letter recognition*	126,739.2373	62	126,892.316	42	-0.12
31	*Optical digits*	140,576.4556	25	140,579.5491	20	-0.002
32	*Shuttle*	1,419,331.0011	45	1,418,934.7769	22	0.02
33	*Person Activitis*	586,257,687,229,319	33	587,431,291,788,442	7	-0.20
34	*Musk clean*	4,634,558.3745	14	4,643,058.9507	9	-0.18
35	*Corel image features*	222,677.0936	110	222,602.8466	33	0.03
36	*Bag of words*	14,466,965,377.3437	81	14,479,992,328.5159	13	-0.09
37	*Covertype*	1,405,197,167.1200	209	1,452,754,545.9700	21	-3.38