TESIS

Universidad de Jaén

Escuela Politécnica Superior de Jaén

DISEÑO Y DESARROLLO DE NUEVOS MODELOS DIFUSOS EVOLUTIVOS PARA AGRUPAMIENTO EN ENTORNOS DE FLUJOS CONTINUOS DE DATOS

Autor:
Luis Alfonso Pérez Martos

Director de la tesis:
Pedro González García y Cristóbal José Carmona del Jesús

Fecha: 15/07/2024

ISBN:

D. Pedro González García y **D. Cristóbal José Carmona del Jesús**, Profesor Titular de Universidad y Catedrático de Universidad respectivamente del Departamento de Informática de la Universidad de Jaén.

CERTIFICAN

Que la memoria titulada "Diseño y desarrollo de nuevos modelos difusos evolutivos para agrupamiento en entornos de flujos continuos de datos" ha sido realizada por D. Luis Alfonso Pérez Martos bajo su supervisión dentro del Programa de Doctorado en Tecnologías de la Información y la Comunicación para optar al grado de doctor.

Jaén, a 13 de julio de 2024

GONZALEZ GARCIA PEDRO - 26210879X

Firmado digitalmente por GONZALEZ GARCIA PEDRO - 26210879X
Fecha: 2024.07.15 09:10:23 +02'00'

Pedro González García
Tutor y Director de la Tesis

CARMONA DEL JESUS CRISTOBAL JOSE - 75102603J

Firmado digitalmente por CARMONA DEL JESUS CRISTOBAL JOSE - 75102603J
Fecha: 2024.07.15 10:08:47 +02'00'

Cristóbal José Carmona del Jesús
Director de la Tesis

A mi familia y mis padres por todo el apoyo incondicional recibido.
Os amo, porque sin vosotros esto no sería posible.

Agradecimientos

Cuando uno empieza a dar los primeros pasos en el mundo de la investigación, suele estar lleno de sueños y expectativas. Siempre pensando en la adquisición de nuevos conocimientos, explorar con innovadoras formas de trabajar y, finalmente, presentando las propuestas desarrolladas. Sin embargo, la realidad de este camino es a menudo mucho más desafiante de lo que uno podría prever.

La investigación es compleja y está lleno de altibajos. En el inicio, la emoción de publicar tus primeros artículos y la curiosidad por descubrir algo nuevo pueden ser muy motivador. Pero a medida que avanza, uno se enfrenta a obstáculos inesperados: la dificultad de encontrar fuentes adecuadas, la ardua tarea de recopilar y analizar datos, el rechazo de artículos y la constante necesidad de revisar y ajustar las hipótesis. Cada paso requiere tiempo, esfuerzo, mucha paciencia y perseverancia. El proceso se vuelve aún más complicado cuando se debe compaginar con otros compromisos importantes, como preparar unas oposiciones. Pero como dice el refranero popular: «Esfuerzo, voluntad y perseverancia son los pilares del éxito.» Por tanto, ahora que estamos dando los últimos pasos de esta compleja tarea que es la investigación, te acuerdas de aquellas personas que han sido participe de este éxito.

En primer lugar, mencionar la labor desempeñada por mi director y tutor Cristóbal y Pedro. La labor de ambos es incomparable, no solo por el conocimiento que han transmitido, sino también por la comprensión y el apoyo incondicional que me han brindado a lo largo de este proceso. Mil gracias por vuestro apoyo.

En segundo lugar, quiero expresar mi agradecimiento a mi esposa Lola. Tu amor, paciencia y comprensión han sido mi mayor fuente de fortaleza. Gracias por estar siempre a mi lado, animándome en los momentos difíciles y celebrando cada pequeño logro conmigo. Tu apoyo inquebrantable ha sido fundamental para la realización de este trabajo.

A mis queridas hijas, Carlota y Lola, les dedico este logro con especial cariño. Ellas son mi mayor inspiración y motivo de orgullo. Gracias por llenarme de alegría y por ser una constante fuente de motivación. Sus sonrisas y abrazos han sido mi refugio y mi impulso para seguir adelante. Esta tesis también es vuestra, porque somos un pedazo de equipo. Os quiero.

Finalmente, a mis padres, por inculcarme el valor del esfuerzo y la dedicación desde una edad temprana. Su confianza en mí ha sido una guía constante.

«Gracias de todo corazón.»

Resumen

Hoy, la extracción de conocimiento de los datos se enfrenta a grandes retos por su crecimiento exponencial en volumen y complejidad provocando un gran desafío para la comunidad científica, ya que los sistemas tradicionales presentan limitaciones. Dentro de la Ciencia de Datos podemos encontrar diversas disciplinas enfocadas en encontrar estructuras o relaciones de interés en datos no relacionados. Esta técnica, conocida como agrupamiento, se emplea en el aprendizaje no supervisado para establecer lazos de unión entre los datos.

Algunas de las actividades más comunes dirigidas a mejorar el agrupamiento se enfocan en la optimización de parámetros, el manejo de datos complejos, la exploración del espacio de soluciones, la búsqueda de óptimos locales y la adaptación dinámica de los datos mediante algoritmos evolutivos inspirados en los principios de la evolución natural. En entornos complejos, la calidad de los grupos puede verse afectada por la posibilidad de que un mismo dato pueda pertenecer a diferentes grupos. Por lo tanto, es fundamental adoptar un enfoque integral al agrupar datos para asegurar la alta calidad de los grupos resultantes. Los datos complejos suelen estar presentes, especialmente, en entornos de *Big Data* y flujos continuos de datos, entre otros. El agrupamiento en entornos de flujos continuos está despertando interés por la posibilidad de extraer conocimiento de forma rápida y en tiempo real sin la necesidad de almacenar los propios datos. Sin embargo, es importante destacar que existe una falta de análisis en el agrupamiento de datos complejos, desde múltiples perspectivas.

A lo largo de esta tesis, profundizaremos en el análisis del agrupamiento y en cómo los algoritmos evolutivos mejoran la optimización y formación de grupos desde diversas perspectivas, especialmente en entornos con datos complejos, co-

mo el *Big Data* y los flujos continuos de datos. Se presentan diversas propuestas para mejorar las soluciones existentes en varios campos. Para el agrupamiento, dada la amplia variedad de algoritmos disponibles una de las contribuciones fundamentales de esta tesis es el desarrollo de un paquete en R que integra los algoritmos jerárquicos y particionales más utilizados de la *Task View*[1] de agrupamiento. Este paquete no solo simplifica la ejecución de estos algoritmos, sino que también integra una serie de herramientas de especial interés. Con ellas, es posible comparar diferentes enfoques y realizar un análisis exhaustivo de los resultados obtenidos. Estas funcionalidades permiten explorar posibles soluciones y seleccionar la más valiosa para el problema planteado. En el ámbito de los algoritmos evolutivos aplicado al agrupamiento, presentamos una propuesta que combina el algoritmo evolutivo CHC con hiper-rectángulos para optimizar los óptimos locales, mejorando la distribución de los datos en grupos respectos a algoritmos tradicionales de agrupamiento. Dada la eficacia de este enfoque, se han implementado mejoras para abordar el problema desde múltiples perspectivas. Se ha refinado la asignación de datos cuando pueden ser agrupados en distintos grupos, empleando una función de *screening* que filtra y selecciona los resultados más relevantes. Además, se introduce una función de consenso que pondera los mejores resultados de la función *screening* para seleccionar el mejor agrupamiento de entre los n posibles. Para finalizar la tesis, se introduce una propuesta destinada a entornos de *Big Data* y flujos continuos de datos, con el objetivo de extraer conocimiento en entornos con datos complejos. Esta propuesta implica evaluar los datos desde diversas perspectivas al agruparlos.

Los resultados de las propuestas ofrecen un valioso conocimiento que mejorará el agrupamiento en entornos complejos por parte de la comunidad investigadora.

[1] https://cran.r-project.org/web/views/Cluster.html

Índice general

Índice de figuras

Glosario

ACO: Algoritmo de optimización por colonia de hormigas.

EA: Algoritmo evolutivo.

FA: Algoritmo de luciérnagas.

KDD: Descubrimiento de conocimiento en bases de datos.

NMI: Información mutua normalizada.

PSO: Algoritmo de optimización por enjambre de partículas.

1 Introducción

En el siglo XXI, nos encontramos inmersos en un proceso de digitalización con el objetivo de mejorar los recursos y tomar decisiones de manera más eficiente y rápida. Este proceso de transformación consiste en la aplicación de recursos y capacidades digitales que tratan de incrementar el valor de estos, provocando un aumento en el volumen de los datos digitales provenientes de diferentes fuentes de información, tales como correos electrónicos, redes sociales, comercio electrónico, internet de las cosas y dispositivos inteligentes entre otros. Un ejemplo del volumen de información que se transmite por minuto en Internet puede ser visualizado en la figura 1.1, la cual muestra la información generada en un minuto por plataformas y dispositivos inteligentes.

El tratamiento de este volumen de información se vuelve complejo para los sistemas tradicionales encargados de su gestión, por lo que surge un nuevo paradigma que trata de procesar grandes volúmenes de información conocido como *Big Data* [Sagiroglu et al., 2013]. El término *Big Data* se caracteriza por el tamaño de los datos, la velocidad con la que se generan y la variedad de estos, que suelen ser muy diversos y heterogéneos, por lo que la importancia de poder relacionar los datos y obtener conocimiento de este implica un proceso de descubrimiento y análisis. La Ciencia de Datos, también conocida como *Data Science*, es un campo interdisciplinar donde convergen disciplinas como Ciencias de la Computación, Estadísticas y Matemáticas. Su objetivo es analizar y obtener información signi-

Figura 1.1: Información transferida por minuto en internet. *Fuente: ediscoverytoday.com/.*

ficativa a partir de grandes conjuntos de datos. El conjunto de disciplinas que componen la Ciencia de Datos queda reflejado en la figura 1.2. Cada una de estas disciplinas emplea una serie de procesos de forma iterativa que se utilizan de manera frecuente para poder extraer reglas o patrones que nos permitan dar respuestas a los problemas planteados. Este proceso se conoce como Descubrimiento de Conocimiento en Bases de Datos (KDD) [Fayyad, 1997].

Figura 1.2: Disciplinas que componen la Ciencia de Datos. *Fuente: elaboración propia.*

KDD se utiliza habitualmente en la comunidad científica en múltiples disciplinas, pero su uso se destaca especialmente en el análisis de grandes volúmenes de datos, donde se combinan las acciones de descubrimiento con análisis para extraer patrones que se usarán para análisis posteriores [Fayyad et al., 1997]. El proceso de KDD para descubrir conocimiento queda expuesto en la figura 1.3 y se organiza en las siguientes fases:

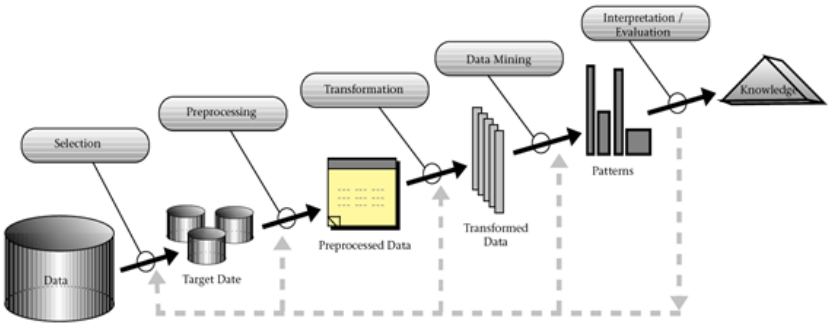


Figura 1.3: Fases del proceso KDD. *Fuente: [Fayyad et al., 1996].*

- Selección: es la fase inicial y consiste en identificar la meta en el descubrimiento de conocimiento. El proceso de identificación se lleva a cabo mediante la selección de un conjunto de datos o muestras. Este proceso de selección puede variar en función de las metas.

- Preprocesamiento/limpieza: esta fase es muy tediosa ya que requiere de mucho tiempo. Consiste en revisar y descartar los datos vacíos, desconocidos, duplicados y aquellos que generan ruido (datos que se encuentran fuera de los límites establecidos para alcanzar las metas), entre otros.

- Transformación/reducción: la transformación consiste en realizar transformaciones sintácticas en los datos, aplicando métodos de reducción de dimensión o transformación para reducir el efecto que provocan las variables de baja importancia.

- Minería de datos: consiste en la búsqueda y descubrimiento de patrones aplicando un conjunto de algoritmos. La elección del algoritmo incluye la selección del método en base al patrón, así como los parámetros del mismo dependiente del tipo de dato.

- Interpretación/evaluación: implica la interpretación de los resultados y su traducción en términos que sean comprensibles para los usuarios.

En el contexto del KDD, la minería de datos es un proceso de descubrimiento y análisis de patrones, tendencias y relaciones significativas en conjuntos de datos grandes y complejos mediante el uso de algoritmos cuya elección va a depender de la meta a alcanzar. Estas metas se componen de dos inducciones bien definidas.

- Descriptiva: el propósito de estas técnicas es identificar patrones que nos permitan describir e interpretar los datos. Normalmente se emplean en el aprendizaje no supervisado y se centran en encontrar estructuras o relaciones que faciliten la extracción de información interesante a partir de datos no etiquetados previamente. Los algoritmos más destacados incluyen:

    - Agrupamiento [MacQueen et al., 1967]: consiste en agrupar los datos en conjuntos donde los elementos compartan similitudes significativas entre sí.
    - Reducción de dimensionalidad [Kasun et al., 2016]: los algoritmos de reducción de dimensión tratan de obtener la información crucial reduciendo el número de características del conjunto de datos.
    - Detención de anomalías [Hawkins, 1980]: son útiles en aquellos casos donde es necesario identificar muestras poco frecuentes o anómalas.
    - Asociación [Agrawal et al., 1993]: los algoritmos de asociación tratan de encontrar patrones o relaciones de interés en el conjunto de datos.

- Predictiva: utilizan el conocimiento de ciertas variables para anticipar valores desconocidos o futuros de otras variables. Se emplea de forma habitual con aprendizaje supervisado y se basa en hacer predicciones o tomar decisiones a partir de datos etiquetados dentro de un conjunto de datos de entrada. Entre las técnicas más comunes del aprendizaje supervisado se destacan:

- Clasificación [Madhusmita Das et al., 2020]: consiste en asignar de manera precisa los datos desconocidos o futuros en categorías específicas.

- Regresión [M. Kim et al., 2020]: es utilizada para establecer la relación entre variables dependientes e independientes.

El aprendizaje automático [Samuel, 1967] (*Machine Learning* en inglés) es una rama de la Inteligencia Artificial que se encarga de desarrollar técnicas que permiten a los ordenadores aprender y mejorar su rendimiento sin necesidad de una programación explícita. Entre las técnicas más extendidas para este aprendizaje encontramos los algoritmos evolutivos [Eiben et al., 2015], basados en algoritmos estocásticos de búsqueda directa en poblaciones que imitan, en cierto sentido, la evolución natural [Eiben et al., 2015]. Estos algoritmos son frecuentemente utilizados para encontrar una solución óptima en problemas complejos. Se basan en la idea de evolucionar una población de posibles soluciones a lo largo de múltiples generaciones para mejorar gradualmente la calidad de las soluciones. En la literatura, estos algoritmos se aplican con frecuencia en la optimización de problemas complejos como por ejemplo en agrupamiento.

El agrupamiento [Rodriguez et al., 2019] [A. Ahmad et al., 2019] se refiere a un proceso mediante el cual se organizan conjuntos de datos en grupos, donde los elementos dentro de cada grupo comparten características similares, mientras que los elementos de diferentes grupos son diferentes entre sí. El propósito fundamental del agrupamiento es desvelar estructuras subyacentes, relaciones y patrones inherentes en los datos, lo cual puede enriquecer la comprensión para una toma de decisiones más informada. Hay diferentes tipos de agrupamiento, siendo los más utilizados: jerárquico, particional, basado en densidad y en centroides, entre otros destacados. Al trabajar con datos no etiquetados, el agrupamiento [Mythili et al., 2014] presenta una serie de desafíos, como la distribución, tamaño, densidad y orientación de los grupos, además de ofrecer una perspectiva única del agrupamiento de los datos.

En la última década ha surgido una corriente que trata de describir y mejorar el agrupamiento de los datos teniendo en cuenta múltiples perspectivas. Esto implica agrupar los datos en grupos con diferentes densidades, orientaciones y tamaños, de manera que el proceso de toma de decisiones sea más completo al permitir describir el agrupamiento desde diferentes puntos de vista. Esta corriente se conoce como multi-agrupamiento. Una diferencia clave entre el agru-

pamiento y el multi-agrupamiento es que el primero utiliza un solo algoritmo para agrupar los datos, mientras que el segundo utiliza múltiples algoritmos y combina los resultados para obtener la mejor solución al problema.

La combinación de las distintas disciplinas de la Ciencia de Datos es esencial en la resolución de problemas, especialmente para datos no etiquetados donde no existe una correlación entre ellos. Este enfoque puede ser especialmente interesante en conceptos como el internet de las cosas, donde disponemos de dispositivos que están continuamente intercambiando y transfiriendo datos de manera continua (conocido en inglés como *Data Streaming*) y donde es necesario obtener patrones que nos permitan relacionar los datos.

Objetivos

En este trabajo de investigación se plantean propuestas de aprendizaje no supervisado para tratar de solucionar problemas complejos en entornos de flujos continuos de datos. El principal objetivo de esta tesis es el diseño y desarrollo de nuevos modelos difusos evolutivos para agrupamiento en entornos de flujos continuos de datos. Este objetivo se desarrolla en los siguientes objetivos específicos:

- Análisis e identificación de los principales enfoques utilizados en el agrupamiento.
- Desarrollo de nuevas propuestas de agrupamiento basadas en sistemas difusos evolutivos.
- Análisis y desarrollo de nuevas propuestas en entornos complejos de multi-agrupamiento de flujos continuos de datos.
- Estudio de las propuestas en entornos reales de flujos continuos de datos y preprocesamiento asociado.

Estructura de la memoria

La estructura de esta tesis se compone de los siguientes capítulos:

- El capítulo 2 introduce el concepto de agrupamiento, incluyendo el análisis de grupos, las medidas de distancia, los tipos de agrupamiento y las métricas de calidad encargadas de evaluar la distribución de los datos en

los grupos, además de un estudio de las principales aportaciones realizadas sobre agrupamiento por la comunidad en R. Al finalizar este capítulo, presentamos una propuesta de un nuevo paquete en R llamado *Clustering*. Este paquete, desarrollado completamente en R, incluye algoritmos avanzados de agrupamiento que permiten evaluar la calidad de los grupos mediante diversas métricas. Además, proporciona herramientas visuales que facilitan la interpretación y comparación de los resultados de los agrupamientos.

- Durante el capítulo 3, se presenta los algoritmos evolutivos como una estrategia de optimización inspirada en la evolución natural. En el contexto del análisis de datos, los algoritmos evolutivos destacan como una herramienta poderosa para resolver una amplia gama de problemas, incluido los problemas de sensibilidad a los parámetros y óptimos locales que tienen los algoritmos de agrupamiento. Finalizamos el capítulo con una amplia variedad de soluciones que ejemplifican la aplicación de los algoritmos evolutivos en diversas áreas de la sociedad.

- Dentro del capítulo 4 se abordan temas clave como la lógica difusa, la minería de flujos de datos y el concepto de hiper-rectángulo, que permiten facilitar la compresión del funcionamiento de los algoritmos propuestos. Dichos algoritmos mejoran el proceso de agrupamiento de datos en comparación con los algoritmos de agrupamiento tradicional. Al final del capítulo, se presenta una nueva propuesta de agrupamiento de datos en entornos donde el agrupamiento tradicional presenta problemas debido a la frecuencia con que los datos son recibidos y deben ser procesados como son los flujos continuos de datos.

- En el capítulo 5 nos adentraremos en un estudio experimental detallado. Este estudio implicará el uso de una variedad de conjuntos de datos cuidadosamente seleccionados con el propósito específico de evidenciar las mejoras proporcionadas por los algoritmos propuestos en comparación con los algoritmos de agrupamiento clásicos. A través de este enfoque experimental, buscamos ofrecer una evaluación objetiva y rigurosa del rendimiento de los algoritmos propuestos en una variedad de escenarios y condiciones. Este análisis experimental nos permitirá no sólo demostrar la eficacia de nuestras propuestas, sino también comprender mejor sus fortalezas y limitaciones en diferentes contextos de aplicación.

- Finalmente, los apéndices incluyen un estudio experimental con la librería *Clustering* y las tablas con los resultados obtenidos por los distintos algoritmos propuestos a lo largo de esta memoria.

2 Agrupamiento

A lo largo de este capítulo, abordaremos algunos de los aspectos fundamentales sobre los cuales se construye esta tesis. Exploraremos los conceptos y fundamentos relacionados con el agrupamiento, incluyendo su definición, tipos, medidas de distancia y criterios de calidad. Analizaremos las librerías en R sobre agrupamiento usadas con más frecuencia y realizaremos una comparativa sobre las funcionalidades incorporadas por estas librerías. Cerraremos este capítulo presentando la librería *Clustering*. La librería *Clustering* contribuye a mejorar la comparación dentro del proceso de agrupamiento de datos con diversas funcionalidades no implementadas hasta el momento. Además, esta librería pone en práctica los diversos conceptos y técnicas tratadas a lo largo de este capítulo y que son esenciales a la hora de crear agrupamientos de calidad.

2.1. Concepto de agrupamiento

El agrupamiento (*clustering* en inglés) es una técnica descriptiva de la minería de datos que busca crear grupos (*clusters*) de un conjunto de datos, con el objetivo de que los datos contenidos dentro de un grupo sean lo más similares entre sí y que la distancia entre los grupos sea la máxima posible. A la hora de agrupar los datos, puede resultar complicado determinar a qué grupo pertenece un dato. Puede darse el caso de que ciertos parámetros o características

sean más adecuados para un tipo de agrupamiento en concreto, e incluso el tamaño del conjunto de datos puede influir en los criterios de agrupamiento. El desafío de la dimensionalidad agrava el problema, ya que no solo afecta al costo computacional, sino también a la estabilidad de los algoritmos. Sin embargo, se proponen métodos de selección de características como solución a este problema [Saxena et al., 2017].

Formalmente, el agrupamiento se define de la siguiente manera:

Definición 1 *Dado un conjunto de datos* $X = x_1, x_2, ..., x_n$*, donde* x_i *es un vector de características en un espacio métrico de* p*-dimensiones y* $N = |X|$ *es el número de datos en* X*; entonces, un agrupamiento válido de* X *es un conjunto de grupos* $C = \{C_1, C_2, \ldots, C_M\}$*, donde* M *es el número de grupos que tienen las siguientes propiedades de partición [Veenman et al., 2003]:*

1. $C_i \neq \emptyset, 1 \leqslant i \leqslant M$
2. $\bigcup_{i=1}^{M} C_i = X$
3. $C_i \cap C_j = \emptyset, \quad i \neq j, \quad 1 \leqslant i, j \leqslant M$

Para particionar el conjunto de datos en grupos cuyos datos sean lo más similares posible, es necesario medir la distancia entre sus datos basándose en la representación de los datos en el espacio de variables. Para calcular estas medidas, es necesario entender el concepto de distancia. La distancia se define como [Ruiz, 2011]:

Definición 2 *Dado un conjunto* X*, la distancia en* X *es una función* $X \times X \to \mathbb{R}^+$ *donde a cada par de elementos* $x, y \in X$ *se le asocia el número* $d_{x,y}$*, que cumple las siguientes propiedades:*

1. $d(x, y) = 0 \Leftrightarrow x = y$
2. $d(x, y) = d(y, x)$
3. $d(x, y) \leqslant d(x, z) + d(y, z)$

Al par (X, d) se le conoce como espacio métrico. Si $|\cdot|$ es una norma en un espacio vectorial E, la función sería de la siguiente forma:

$$d(x, y) := |x - y| \tag{2.1}$$

lo cual define una distancia en E. Por lo tanto, cada espacio normado es un espacio métrico. En el espacio euclidiano n-dimensional $\mathbb{R}^q$, la distancia se define como:

$$d_e(x, y) = \left[\sum_{i=1}^{n}(x_i - y_i)^2\right]^{\frac{1}{2}} \tag{2.2}$$

En la literatura podemos encontrar una gran variedad de revisiones sobre agrupamiento como por ejemplo [S. Singh et al., 2020], [Y. Zhang et al., 2019], donde se explican los conceptos básicos y los principales algoritmos utilizados para agrupar conjuntos de datos. [Saxena et al., 2017] presenta un estudio comparativo sobre diversas medidas de evaluación y criterios de agrupamiento. [Ezugwu, 2020] realiza una revisión completa sobre los algoritmos metaheurísticos inspirados en la naturaleza aplicados en el análisis automático del agrupamiento.

Los algoritmos de agrupamiento pueden clasificarse en particiones duras (*hard*) y suaves (*soft*). En las particiones duras, cada dato se asigna exclusivamente a un solo grupo, sin superposiciones. Uno de los algoritmos más conocidos es k-means [Sinaga et al., 2020], que tiene como objetivo minimizar la suma de las distancias al cuadrado entre los datos y los centroides de sus respectivos grupos. Por tanto, siendo C_i el conjunto de datos asignados al grupo i y μ_i el centroide del grupo i, el cálculo de la suma de las distancias al cuadrado se define como:

$$C = \sum_{i=1}^{K}\sum_{j \in C_i} |x_j - \mu_i|^2 \tag{2.3}$$

donde K es el número de grupos, x_j representa un dato, μ_i es el centroide del grupo i, y $|x_j - \mu_i|$ representa la suma de las distancias entre todos los datos de cada grupo y su respectivo centroide.

El particionamiento suave, también conocido como particionamiento difuso [Ruspini et al., 2019], es un tipo de agrupamiento en el que a cada dato del conjunto se le asigna un grado de pertenencia a múltiples grupos en lugar de asignarlo exclusivamente a uno solo, como ocurre en el agrupamiento duro. Al tratarse de asignaciones probabilísticas de los datos a los grupos, refleja la incertidumbre o ambigüedad asociada a los datos. Su característica principal es la representación de la pertenencia parcial de los datos a diferentes grupos. Uno de los métodos más utilizados es el fuzzy c-means [James C Bezdek et al., 1984]. En

el fuzzy c-means, cada dato se asocia con valores de pertenencia a cada grupo, y estos valores suman 1. El objetivo del algoritmo es minimizar la incertidumbre general en la asignación de grupos.

2.2. Análisis de grupos

El análisis de grupos (*cluster analysis* en inglés) busca crear grupos homogéneos de datos. Dado que las soluciones no son únicas debido a que la asignación a un grupo u otro depende de los parámetros seleccionados, se lleva a cabo un proceso de análisis que consta de varias etapas [Backhaus et al., 2021].

Las etapas del análisis de grupos se esquematizan en la figura 2.1 y se detallan a continuación:

Figura 2.1: Etapas que componen el análisis de grupos. *Fuente: elaboración propia.*

Variables [Řezanková et al., 2009]: seleccionar las características para describir cada dato constituye una pieza esencial en el proceso de análisis de grupos. Es crucial que las variables seleccionadas sean relevantes para el agrupamiento buscado. La elección inicial de variables actúa como una categorización preliminar de los datos, con limitadas directrices matemáticas y estadísticas. También es importante considerar el número de variables: un exceso de ellas puede complicar el análisis y añadir ruido a la estructura de los grupos. Las variables pueden ser de diferentes tipos (categóricas, ordinales e intervalo), lo que requiere una normalización previa al análisis. Para variables de tipo intervalo, se suele tipificarlas calculando las desviaciones típicas. Sin embargo, esto puede diluir diferencias significativas, por lo que algunos autores sugieren usar la desviación estándar entre grupos. Cuando las variables son de distintos tipos, convertirlas en binarias antes del cálculo de similitudes es una técnica común y efectiva, pero tiene la desventaja que puede provocar pérdida de información. Asimismo, para variables mixtas, se puede realizar un análisis separado para cada tipo de variable y luego sintetizar los resultados obtenidos de los diferentes análisis.

Medidas de distancia [Landau et al., 2010]: un gran número de métodos de agrupamiento requieren una medida de asociación para evaluar la proximidad entre los datos de un conjunto de datos. Para medir la proximidad entre los datos, generalmente se mide en términos de distancias. Existe una gran variedad de medidas y la elección depende de las variables y el objetivo del agrupamiento. Las medidas de distancia tratan de minimizar la distancia entre los datos. Algunas de las medidas más conocidas son: Euclídea [Irani et al., 2016], Manhattan [Archana Singh et al., 2013], Correlación de Pearson [Berthold et al., 2016], Coeficiente de Jaccard [Bag et al., 2019], Hamming [Norouzi et al., 2012] y Mahalanobis [Sitaram et al., 2015], entre otras. Estas medidas se describen con más detalle en las secciones siguientes.

Métodos de agrupamiento [Brian Everitt, 1980]: el número de propuestas de agrupamiento ha ido creciendo en los últimos tiempos, dividiéndose de forma general en jerárquicos y no jerárquicos.

- Agrupamiento jerárquico: trata de fusionar grupos para formar uno nuevo, o dividir uno existente en dos. Este proceso de fusión o división se realiza de forma iterativa, hasta maximizar o minimizar una medida de distancia.

- Agrupamiento no jerárquico: en este tipo de agrupamiento se reemplaza la noción de enlaces por la idea de definir la forma de los grupos. Se centra más en determinar y definir los grupos en términos de asignaciones de puntos a grupos, posiciones de centroides o medoides, o en la estructura de densidad en el espacio de características. Es necesario indicar el número de particiones. Algunos de los métodos más comunes son: K-means, K-Medoids [Ushakov et al., 2021] y Dbscan [Deng, 2020].

Número de grupos [Löster, 2016]: la elección del número de grupos es una parte esencial en el agrupamiento ya que afecta a la calidad de los resultados. Algunas de las consideraciones para obtener un número óptimo son mencionadas a continuación:

- Método del codo (*Elbow Method* en inglés) [Bholowalia et al., 2014]: consiste en ejecutar el algoritmo de agrupamiento con diferentes valores de k y representar de forma gráfica la variación de las sumas de los cuadrados en función de k hasta detectar que la inercia comience a disminuir de forma menos significativa formando un codo.

- Coeficiente de silueta (*Silhouette* en inglés) [Starczewski et al., 2015]: consiste en calcular la calidad de los grupos para diferentes valores de k y seleccionar el k que maximiza el valor medio de este coeficiente.

- Densidad: basado en el criterio de densidad, como en el caso de `Dbscan`, analiza la densidad de los puntos y la conectividad para determinar la cantidad de grupos.

Validación [Halkidi et al., 2001]: en esta etapa, se obtienen conclusiones sobre los grupos generados y se lleva a cabo una evaluación exhaustiva de su confiabilidad y validez. Para validar los resultados del agrupamiento, se sugiere seguir estos procedimientos:

- Ejecutar el algoritmo con el mismo conjunto de datos utilizando diferentes medidas de distancia para buscar la solución más estable.

- Emplear diversos algoritmos de agrupamiento y comparar sus resultados.

- Analizar los resultados del agrupamiento en subconjuntos de datos donde se reduzca el número de atributos, y validar los resultados en función del análisis completo.

- En el caso del agrupamiento no jerárquico, dado que depende del orden de los casos en el conjunto de datos, se recomienda variar el orden hasta encontrar una solución estable.

En las siguientes secciones, se presentará un análisis de las etapas que componen el análisis de grupos. Inicialmente, se explorarán las diversas medidas de distancia, que son esenciales para cuantificar las similitudes y diferencias entre los datos. A continuación, se profundizará en los distintos métodos de agrupamiento, detallando sus características. Finalmente, se abordará la validación de la calidad de los grupos, utilizando diversas métricas.

2.3. Medidas de distancia

Las medidas de proximidad o distancia en el agrupamiento cuantifican la cercanía entre los datos en un conjunto de datos, ya sea de manera explícita o implícita, revelando el nivel de conexión entre ellos. La relación de agrupación entre los datos es fundamental en todos los algoritmos de agrupamiento, y su

eficacia depende en gran medida del método utilizado. A continuación, ofreceremos un breve resumen de las medidas de distancia más utilizadas tanto en el agrupamiento tradicional como en las propuestas más recientes.

2.3.1. Distancia Euclídea

La distancia Euclídea (*Euclidean* en inglés) para los puntos x, y en un espacio n-dimensional se define (Ecuación 2.4) de la siguiente manera:

$$d_e(x, y) = \left[\sum_{i=1}^{n} (x_i - y_i)^2 \right]^{\frac{1}{2}} \tag{2.4}$$

2.3.2. Distancia de Manhattan

La distancia de Manhattan (o métrica del taxista) (Ecuación 2.5) recibe este nombre porque se asemeja a la forma en la que un taxista se movería en una ciudad como Manhattan (Nueva York), al circular por calles cuadriculadas donde solo está permitido movimientos en ángulos rectos. La forma de calcular la distancia es mediante la suma del valor absoluto de la diferencia entre partes. Para el supuesto de dos puntos x, y en un espacio n-dimensional sería el siguiente:

$$d_m(x, y) = \sum_{i=1}^{n} |x_i - y_i| \tag{2.5}$$

2.3.3. Distancia de Chebyshew

También conocida como distancia infinita o distancia del máximo valor absoluto (Ecuación 2.6), mide la distancia absoluta máxima entre todos los datos, por lo que en el caso de dos puntos x, y la distancia infinita o máxima se obtiene de la siguiente forma:

$$d_c = \max_{K=1}^{n} |x_k - y_k| \tag{2.6}$$

2.3.4. Distancia de Minkowski

Para dos puntos x, y en un espacio n-dimensional, la distancia de Minkowski (Ecuación 2.7) se define como:

$$d_m = \left[\sum_{k=1}^{n} |x_k - y_k|^2\right]^{\frac{1}{r}}, r \geqslant 1 \tag{2.7}$$

donde r define el orden de las distancias. Minkowski reduce a la distancia de Manhattan cuando $r = 1$ y a la distancia Euclídea cuando $r = 2$ tal y como se refleja en la (Ecuación 2.8). Diferentes valores de r permiten ajustar la sensibilidad de la medida de distancia a diferentes características de los datos. Para el caso de $r \to \infty$ se determina que:

$$\lim_{x\to\infty} \left[\sum_{k=1}^{n} |x_k - y_k|^r\right]^{\frac{1}{r}} = \max_{k=1}^{n} |x_k - y_k| \tag{2.8}$$

2.3.5. Distancia mediana Euclídea

Es una variación de la distancia de Euclídea (Ecuación 2.9) en la que dos puntos que no comparten ningún atributo en común nunca tendrán una distancia más pequeña que otro par de puntos que contenga los mismos valores. Para el caso de los puntos x e y se obtiene:

$$d_{me}(x, y) = \left[\frac{1}{n}\sum_{k=1}^{n} (x_k - y_k)^2\right]^{\frac{1}{2}} \tag{2.9}$$

2.3.6. Distancia de Chord

Este tipo de medida se utiliza comúnmente en teoría de grafos y geometría (Ecuación 2.10) para calcular la distancia entre dos puntos en un círculo o circunferencia. Se formula de la siguiente forma:

$$d_{cho}(x, y) = \left[2 - 2\frac{\sum_{k=1}^{n} x_k y_k}{\|x\|_2 \|y\|_2}\right]^{\frac{1}{2}} \tag{2.10}$$

donde $\|\cdot\|_2$ es la norma L_2 (Ecuación 2.11) y (Ecuación 2.12):

$$\|x\|_2 = \sqrt{\sum_{k=1}^{n} x_k^2} \tag{2.11}$$

$$\|y\|_2 = \sqrt{\sum_{k=1}^{n} y_k^2} \tag{2.12}$$

2.3.7. Distancia Geodésica

Es una transformación de la distancia de Chord y es la distancia más corta entre puntos en una superficie curva (Ecuación 2.13). El cálculo de esta distancia se lleva a cabo de la siguiente manera:

$$d_g = \arccos\left(1 - \frac{d_{cho}(x,y)}{2}\right) \tag{2.13}$$

2.3.8. Distancia de Mahalanobis

La distancia de Mahalanobis (Ecuación 2.14) se calcula como:

$$d_{ma}(x,y) = \sqrt{(x-y)^{\Sigma-1}(x-y)^{\tau}} \tag{2.14}$$

donde Σ representa la matriz de covarianzas n x n (Ecuación 2.15) donde el elemento (r,s) es la covarianza entre las variables x_r, y x_s (Ecuaciones 2.16 y 2.17), la cual se calcula de la siguiente forma:

$$\Sigma = \begin{pmatrix} c_{11} & c_{12} & \cdots & c_{1q} \\ c_{21} & c_{22} & \cdots & c_{2q} \\ \vdots & \vdots & \ddots & \vdots \\ c_{q1} & c_{q2} & \cdots & c_{qq} \end{pmatrix} \tag{2.15}$$

$$c_{rs} = \frac{1}{n}\sum_{i=1}^{n}(x_{ir} - \overline{x}_r)(x_{is} - \overline{x}_s) \tag{2.16}$$

$$\overline{c}_k = \frac{1}{n}\sum_{i=1}^{n} x_{ik}, k = 1, 2, ..., n \tag{2.17}$$

requiere un elevado coste computacional para el cálculo de la matriz de covarianza para los n elementos del conjunto de datos.

2.3.9. Métrica de Canberra

La métrica de Canberra (Ecuación 2.18) se utiliza de forma frecuente para analizar datos y estadísticas. Es útil para trabajar con datos que representan frecuencias o proporciones. El cálculo de Canberra para los puntos x e y en un espacio n-dimensional se obtiene:

$$d_{ca}(x, y) = \sum_{k=1}^{n} \frac{|x_k - y_k|}{|x_k| + |y_k|} \tag{2.18}$$

2.3.10. Coeficiente de Czekanowski

También conocido como índice de Czekanowski (Ecuación 2.19) propuesta por Johnson y Wichern (2014) para el cálculo de dos puntos en un espacio n-dimensional de la forma siguiente:

$$d_{cz}(x, y) = 1 - \frac{2 \sum_{k=1}^{n} \text{mín}(x_k, y_k)}{\sum_{k=1}^{n} (x_k + y_k)} \tag{2.19}$$

2.4. Tipos de agrupamiento

El número de consideraciones que hay que tener en cuenta a la hora de elegir una estrategia de agrupamiento de datos hace que la elección de la misma sea compleja. En la literatura [Nanda et al., 2014], [Hancer et al., 2017], [D. Xu et al., 2015] es posible encontrar diferentes enfoques que resuelven este problema basado en el funcionamiento de algoritmos.

A continuación, vamos a presentar algunas estrategias para el agrupamiento de datos, así como los algoritmos más representativos para cada una de ellas.

2.4.1. Agrupamiento jerárquico

Crea un desglose jerárquico de los datos en un dendograma que divide recursivamente el conjunto de datos en grupos cada vez más pequeños. Puede crearse de dos formas: ascendente o descendente [Jain et al., 1999]. El método ascendente, se conoce como aglomerativo y los datos se van combinando sucesivamente en función de las medidas, hasta que todos se unan en uno o cumplan una condición de finalización. El método descendente, se conoce como divisivo, donde todos los datos están en el mismo grupo, y a medida que iteramos se van

dividiendo en subconjuntos más pequeños hasta que cada dato esté en un grupo individual o cumpla una condición de parada. Se pueden utilizar diferentes estrategias para determinar la distancia entre los grupos [F. Nielsen et al., 2016]:

- Enlace completo: la distancia entre dos grupos se mide calculando los datos más lejanos de cada grupo (Ecuación 2.20):

$$d(C_i, C_j) = \max_{x \in C_i, y \in C_j} dist(x, y) \tag{2.20}$$

  donde x e y son los datos más lejanos de los grupos C_i y C_j, respectivamente. Este método tiende a producir grupos con formas esféricas y puede ser sensible a valores atípicos.

- Enlace simple: la distancia entre dos grupos se mide calculando los datos más cercanos de cada grupo (Ecuación 2.21):

$$d(C_i, C_j) = \min_{x \in C_i, y \in C_j} dist(x, y) \tag{2.21}$$

  donde x e y son los datos más cercanos de los grupos C_i y C_j, respectivamente. Este método tiende a producir grupos alargados y puede ser sensible a ruido.

- Enlace promedio: la distancia entre dos grupos se mide calculando la distancia promedio de los datos de cada grupo (Ecuación 2.22):

$$d(C_i, C_j) = \frac{1}{\|C_i\| \cdot \|C_j\|} \sum_{x \in C_i, y \in C_j} dist(x, y) \tag{2.22}$$

  donde x e y son los datos más cercanos de los grupos C_i y C_j, respectivamente. Este método produce grupos más equilibrados y es menos sensible a valores anómalos en comparación con el enlace completo.

- Enlace centroide: mide la distancia entre los centroides de dos grupos C_i y C_j (Ecuación 2.23):

$$d(C_i, C_j) = dist(centroide(C_i), centroide(C_j)) \tag{2.23}$$

  donde x e y son los datos más cercanos de los grupos C_i y C_j, respectivamente. Este enlace puede generar grupos con forma de esfera y es sensible a la variación en el tamaño de los grupos.

Un ejemplo de agrupamiento jerárquico se puede encontrar en la figura 2.2.

Los algoritmos de agrupamiento jerárquico más conocidos son: Cure [Guha et al., 1998], Chameleon [Guo et al., 2019] y Birch [T. Zhang et al., 1996].

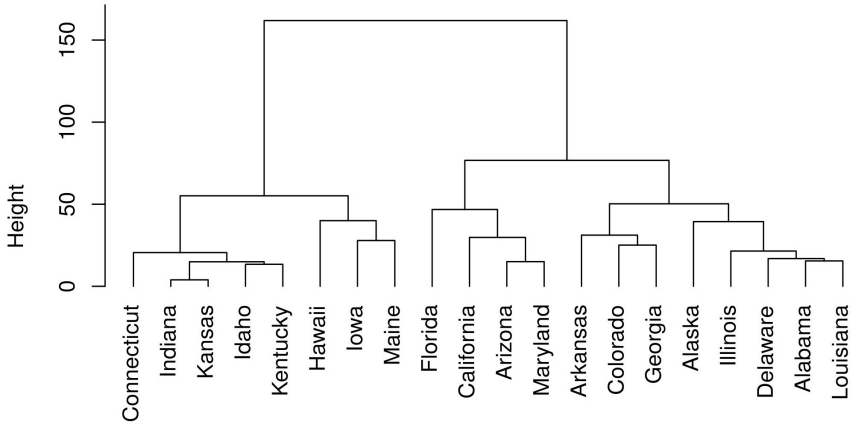


Figura 2.2: Agrupamiento jerárquico de los datos en modo descendente. *Fuente: elaboración propia.*

2.4.2. Agrupamiento particional

Se considera la estrategia de agrupamiento más popular. El objetivo de esta estrategia es reubicar los datos en grupos hasta alcanzar una partición óptima. Divide los datos en k particiones, donde cada partición representa un grupo. El agrupamiento particional organiza los datos dentro de los k grupos de forma que la distancia de cada dato respecto al centro de su grupo o respecto a una distribución de grupos sea mínima. La desviación de un dato dado puede evaluarse de diferentes formas según el algoritmo, y se conoce como función de distancia. Un ejemplo gráfico del funcionamiento del agrupamiento particional se puede ver en la figura 2.3.

Los algoritmos de agrupamiento particional más destacados son: Clarans [Ng et al., 2002], Clara [Ramprasanth et al., 2019], K-prototipo [Nithya et al., 2019], K-modo [Z. Huang, 1997] y K-means [MacQueen et al., 1967].

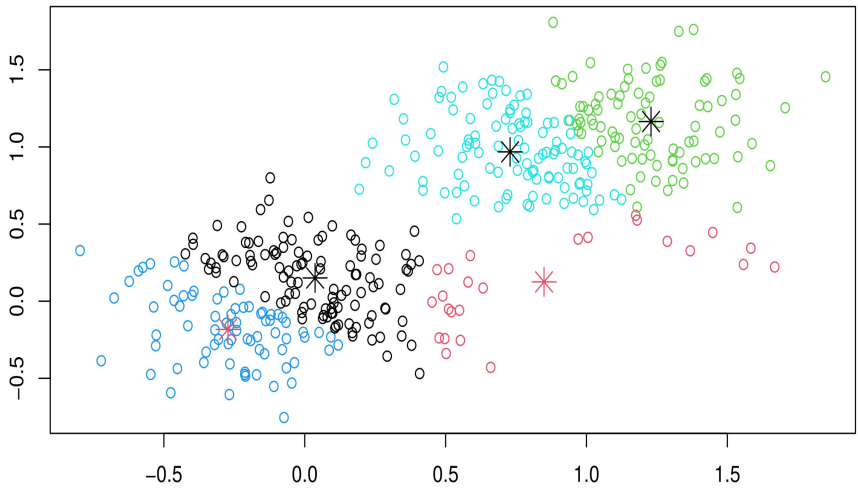


Figura 2.3: Agrupamiento particional con $k = 5$ particiones utilizando k-means. *Fuente: elaboración propia.*

2.4.3. Agrupamiento basado en densidad

Crean grupos basados en regiones densas de datos en el espacio separadas por regiones de baja densidad. La representación gráfica de grupos basados en regiones densas se puede visualizar en la figura 2.4.

Destacamos los siguientes algoritmos: Dbscan [K. Khan et al., 2014], y Denclue [M. Khader et al., 2019].

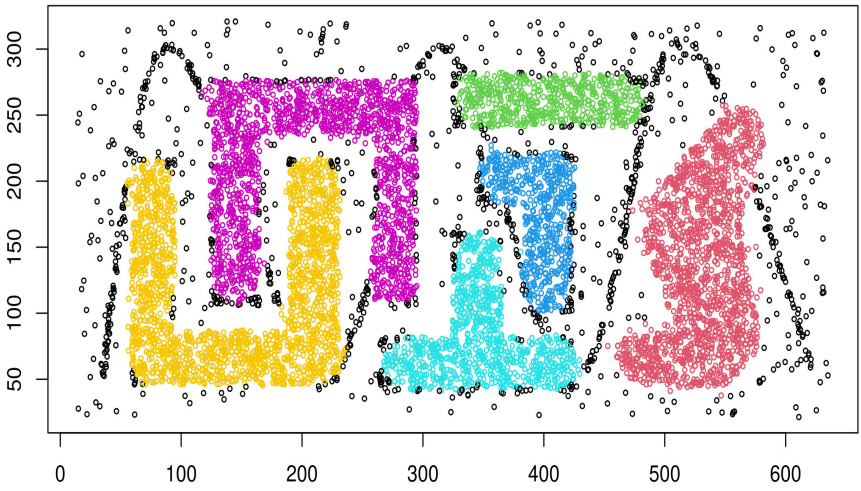


Figura 2.4: Agrupamiento basado en densidad usando el algoritmo Dbscan. *Fuente: elaboración propia.*

2.4.4. Agrupamiento basado en cuadrículas

Es una variación del agrupamiento basado en densidad. Su funcionamiento consiste en dividir el espacio de atributos en una cuadrícula o malla de celdas y asignar datos a las celdas basándose en sus ubicaciones. Las celdas que contienen más de un determinado número de datos se tratan como densas, y se conectan para formar los grupos. La figura 2.5 representa un agrupamiento en cuadrículas.

Algunos de los algoritmos de agrupación basados en cuadrículas más representativos son: Sting [E. W. Ma et al., 2004], Wave Cluster [Mao et al., 2021] y Clique [Rani et al., 2017].

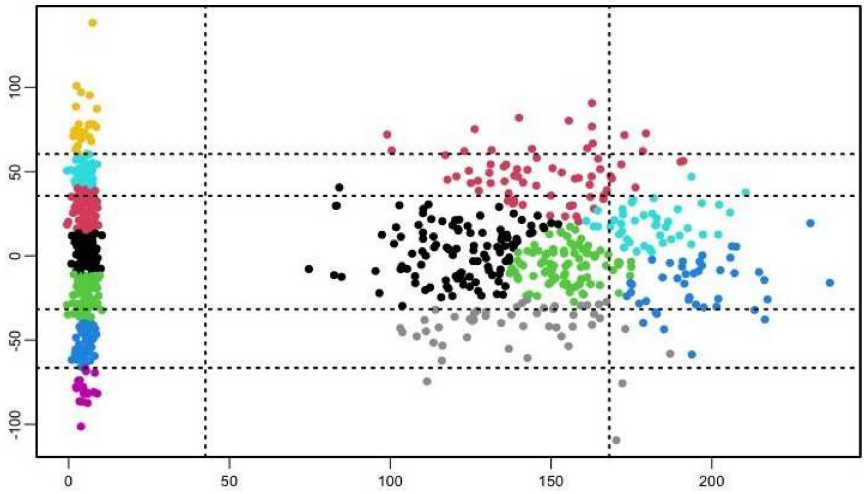


Figura 2.5: Agrupamiento basado en cuadrículas. *Fuente: elaboración propia.*

2.4.5. Agrupamiento basado en modelos

Es un enfoque estadístico de agrupación de datos. Se basa en modelos estadísticos para describir la estructura subyacente de los datos. La idea principal de la agrupación basada en modelos es suponer que los datos se generan a partir de una mezcla de distribuciones de probabilidad, cada una de las cuales corresponde a un grupo diferente. El algoritmo Auto Class utiliza el enfoque bayesiano, partiendo de una inicialización aleatoria de los parámetros que se va ajustando gradualmente para encontrar las estimaciones de máxima probabilidad. En la figura 2.6 tenemos un ejemplo de agrupamiento basado en modelo.

Entre los algoritmos basados en modelos más utilizados destaca SOM [Thalamuthu et al., 2006].

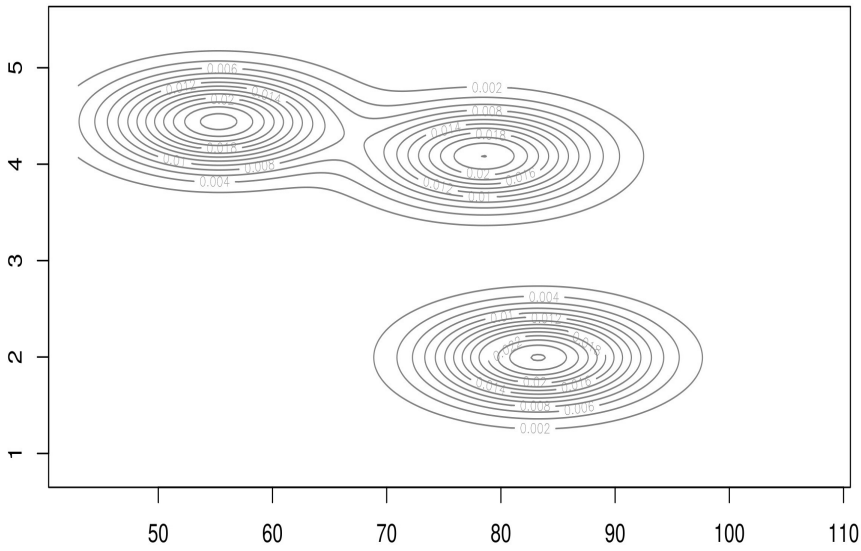


Figura 2.6: Agrupamiento basado en modelos. *Fuente: elaboración propia.*

2.4.6. Agrupamiento basado en grafos

En dicho agrupamiento los datos son representados en grafos, donde cada nodo del grafo representa un dato del conjunto, y una arista es el enlace entre los nodos. Cada arista tiene un coste que se corresponde con la distancia entre dos nodos, según la medida de distancia elegida. La figura 2.7 representa un agrupamiento basado en grafos.

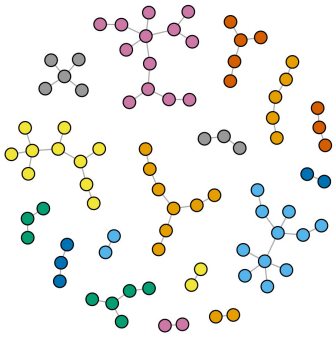


Figura 2.7: Agrupamiento basado en grafos. *Fuente: elaboración propia.*

2.5. Validación: medidas de calidad

A la hora de validar la calidad de los agrupamientos, [Rendón et al., 2011] propone distintos enfoques:

2.5.1. Medidas externas

Están basadas en el conocimiento previo de los datos, en particular su estructura subyacente y el número de grupos que contiene. La idea básica es hacer coincidir el resultado de la partición con la estructura predefinida del conjunto de datos. Como medidas de calidad que son referenciadas de forma frecuente en la literatura destacamos las siguientes:

- Entropía (*Entropy* en inglés) [H. Kim et al., 2007]: evalúa la distribución de las clases en los grupos. Si los grupos están formados por datos de una sola clase la entropía es 0. Cuanto más variada sean las clases dentro de los grupos mejor será la entropía. El valor máximo y mínimo que puede tomar entropía es $[0, \log_b(k)]$, siendo k el número de grupos y b es la base del logaritmo utilizado. La forma de calcular la entropía se puede ver en (Ecuación 2.24):

$$E(j) = \sum_{j=1}^{m} \frac{|C_j|}{k} E_j \tag{2.24}$$

donde C_j es el tamaño del grupo j, k es el número de grupos y m es el número total de datos. Para calcular la entropía de un conjunto de datos (Ecuación 2.25) tenemos que calcular la distribución de clases de los datos en cada grupo de la siguiente manera:

$$E_j = \sum_i p_{ij} \log(p_{ij}) \tag{2.25}$$

donde p_{ij} es la probabilidad de un dato en el grupo i de ser clasificado como clase j.

- Recuperación (*Recall* en inglés) [Abualigah et al., 2018]: mide el número de datos que pertenecen a un grupo en particular y que han sido correctamente agrupados en ese grupo en relación con el número total de datos que realmente pertenecen a ese grupo. Los valores límite que puede tomar recuperación son $[0, 1]$, y la forma de calcularla es (Ecuación 2.26):

$$R(i, j) = \frac{n_{ij}}{n_i} \tag{2.26}$$

  n_{ij} es el número de datos de la clase i que están en el grupo j y n_i es el número de datos en el grupo i.

- Precisión (*Precision* en inglés) [Abualigah et al., 2018]: se centra en medir cuántos de los elementos identificados como pertenecientes a un grupo realmente pertenecen a ese grupo. Los valores máximos y mínimos resultantes deben estar en el mismo rango que para recuperación. La forma de calcular la precisión es (Ecuación 2.27):

$$P(i, j) = \frac{n_{ij}}{n_j} \tag{2.27}$$

  n_j es el número de datos en el grupo j.

- Valor-F (*F-measure* en inglés): busca un equilibrio entre precisión y recuperación, dos métricas consideradas contrapuestas. Los valores resultantes de valor-f deben estar dentro del mismo rango que el indicado para precisión y recuperación. Calcular valor-f se realiza de la siguiente manera (Ecuación 2.28):

$$FM(i,j) = \frac{2 * (P(i,j) * R(i,j))}{(P(i,j) + R(i,j))} \tag{2.28}$$

- Índice Fowlkes-Mallows (*Fowlkes-Mallows Index* en inglés): se encarga de medir la distancia entre dos conjuntos de grupos, con el objetivo de determinar cuántos pares de datos están en el mismo grupo o en grupos diferentes en dos particiones diferentes, obteniendo como resultado la distancia entre dos particiones (Ecuación 2.29). Los valores resultantes admitidos son $[0, 1]$, donde 1 indica un alto grado de similitud y 0 todo lo contrario.

$$FOW(i,j) = \sqrt{P(i,j) * R(i,j)} \tag{2.29}$$

- Variación de la información (*Variation Information* en inglés) [Meilă, 2005]: es utilizada para medir las variaciones o diferencias entre dos agrupamientos diferentes de un conjunto de datos, es decir, se encarga de cuantificar cuanta información se comparte o diferencia entre dos particiones de datos. El cálculo de variación de la información se realiza de la siguiente manera (Ecuación 2.30):

$$VI(i,j) = 2E(i,j) - E(i) - E(j) \tag{2.30}$$

  donde $E(i,j)$ es la unión de la entropía de dos grupos, $E(i)$ es la entropía de i y $E(j)$ es la entropía de j. Los valores admitidos tras el cálculo de la variación de la información deben estar en el rango $[0, \infty)$, donde 0 refleja una gran similitud.

- Índice de Rand (*Rand Index* en inglés) [Chacón et al., 2023]: se encarga de medir la distancia entre dos agrupamientos como la proporción de pares de datos que son asignados al mismo grupo en dos agrupamientos o a grupos diferentes en ambos agrupamientos. Índice de Rand se define como (Ecuación 2.31):

$$RI(a,b) = \frac{a+b}{\binom{n}{2}} \tag{2.31}$$

donde a es el número de pares de datos que están en el mismo grupo en dos agrupamientos, b es el número de pares de datos que están en grupos diferentes en dos agrupamientos y n es el número total de datos que compone el conjunto de datos. Los valores del índice de Rand pueden variar en el rango de $[0, 1]$, donde 1 indica un alto grado de similitud.

- Índice ajustado de Rand (*Adjusted Rand Index* en inglés) [Chacón et al., 2023]: mide la distancia entre dos agrupamientos, mejorando los resultados del índice de Rand ya que a menudo devuelve valores relativamente altos simplemente por azar, especialmente cuando hay un gran número de grupos o cuando hay desequilibrio de datos. Los límites de los valores oscilan en el rango $[-1, 1]$, donde -1 refleja que la similitud es menor de lo esperada. La forma de calcular el índice ajustado de Rand es de la siguiente manera (Ecuación 2.32):

$$ARI(i,j) = \frac{RI(i,j) - E\left[RI(i,j)\right]}{max\left(RI(i,j)\right) - E\left[RI(i,j)\right]} \tag{2.32}$$

- Coeficiente de Jaccard (*Jaccard Coeficient* en inglés): conocido como coeficiente de distancia de Jaccard, es utilizado para comparar la distancia y diversidad entre dos grupos. Es muy similar al índice de Rand, sin embargo, no tiene en cuenta aquellos pares de datos que se encuentran en diferentes grupos para agrupamientos diferentes. Para calcular el coeficiente de Jaccard se realiza de la siguiente manera (Ecuación 2.33):

$$J\left(C, C'\right) = \frac{|C \cap C'|}{|C \cup C'|} \tag{2.33}$$

donde C y C' representa los datos en los grupos. $|C \cap C'|$ es el número de datos compartidos entre ambos grupos. $|C \cup C'|$ es el número de datos totales que son únicos en los grupos. El resultado de calcular el coeficiente de Jaccard debe estar en el rango $[0, 1]$, donde 0 refleja una similitud de poca importancia.

- Información mutua normalizada (*Normalized Mutual Information* en inglés): es una métrica de calidad que cuantifica la dependencia mutua entre dos variables aleatorias basándose en parámetros bien establecidos de la teoría de la información. La NMI se define como (Ecuación 2.34):

$$NMI(C, C') = \frac{I(C, C')}{\sqrt{|E(C), E(C')|}} \tag{2.34}$$

donde C es una variable aleatoria que indica la asignación de los datos en los grupos, y C' es la variable aleatoria que denota las clases subyacente en los datos. $I(C, C') = E(C) - E(C|C')$ es la información mutua entre las variables aleatorias C y C'. $C'(C)$ es la entropía de C. $E(C|C')$ es la entropía condicional de C dado C'. Los valores límites de NMI se sitúan entre $[0, 1]$, donde 0 refleja la ausencia completa de relación y 1 indica la máxima asociación posible entre dos particiones de datos, respectivamente.

- Coeficiente de Mirkin (*Mirkin Metric* en inglés) [S. Wagner et al., 2007]: también conocida como distancia de desajuste de equivalencia (Ecuación 2.35), es una variación del índice de Rand y es muy sensible al tamaño de los grupos:

$$M(C, C') = n(n-1)(1 - R(C, C')) \tag{2.35}$$

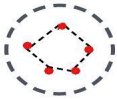
donde n es el número de datos y $R(C, C')$ es el índice de Rand para los grupos. Los valores límites del coeficiente de Mirkin se encuentran en el rango $[0, \infty)$, donde 0 indica una similitud perfecta entre los grupos, mientras que valores mayores sugieren una discrepancia creciente entre ellos.

2.5.2. Medidas internas

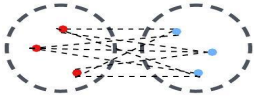
Las métricas internas están basadas en la información intrínseca de los datos. Los conceptos de compactación y separación son fundamentales para medir la calidad interna de los grupos y están representados en la figura 2.8.

- Compactación: se centra en que los datos de cada grupo estén lo más cerca posibles unos de otros. En la figura 2.8a se presenta una representación visual de la compactación entre los datos de un grupo.
- Separación: mide que los grupos estén lo más separados posible unos de otros. En la figura 2.8b se muestra de forma visual la separación entre los datos que pertenecen a diferentes grupos.

  En la figura 2.9 se presentan tres métodos para medir la distancia entre los grupos:



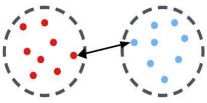
(a) Compactación entre los datos del mismo grupo. *Fuente: elaboración propia.*



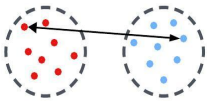
(b) Separación de los datos de dos grupos. *Fuente: elaboración propia.*

Figura 2.8: Representación gráfica de los conceptos de compactación y separación. *Fuente: elaboración propia.*

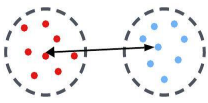
- Vinculación simple: mide la distancia entre los datos más cercanos de los grupos, tal y como viene reflejado en la figura 2.9a.
- Vinculación completa: mide la distancia entre los datos más separados, como se observa en la figura 2.9b.
- Comparación de centroides: mide la distancia entre los centros de los grupos. Gráficamente queda representada esta comparación en la figura 2.9c.



(a) Vinculación simple entre los datos de dos grupos. *Fuente: elaboración propia.*



(b) Vinculación completa entre los datos de dos grupos. *Fuente: elaboración propia.*



(c) Comparación de los centroides entre los datos de dos grupos. *Fuente: elaboración propia.*

Figura 2.9: Comparación de diferentes tipos de vinculación entre datos de dos grupos. *Fuente: elaboración propia.*

A continuación, presentamos un análisis detallado de las métricas internas utilizadas para evaluar los agrupamientos en términos de compactación (que la distancia de los datos dentro del grupo sea la mínima) y separación (que la distancia entre los grupos sea la adecuada). Estas métricas son fundamentales para determinar la efectividad del proceso de agrupamiento y para garantizar que los grupos formados sean coherentes y bien definidos.

- Conectividad (*Connectivity* en inglés) [Deborah et al., 2010]: medida que refleja el grado de relación que hay entre los datos ubicados en el mismo grupo con respecto a sus vecinos más cercanos. Conectividad se puede calcular como (Ecuación 2.36):

$$C = min_{1 \leqslant i \leqslant K} \left(min_{1 \leqslant j \leqslant K, i \neq j} \left(\frac{dist(C_i, C_j)}{max_{1 \leqslant k \leqslant K}\{diam(C_k)\}} \right) \right) \quad (2.36)$$

  donde $dist(C_i, C_j)$ es la distancia entre dos grupos y $diam(C_k)$ es el diámetro de un grupo concreto. Los valores extremos que puede tomar la conectividad se encuentran en el rango $[0, \infty)$, donde 0 indica una compactación mínima y valores mayores representan una mayor compactación entre los grupos.

- Dunn [James C Bezdek et al., 1995]: es una medida que trata de encontrar grupos con la máxima distancia entre ellos (separación) y la mínima distancia entre los datos dentro de cada grupo (compactación). Los valores de Dunn se encuentran en el rango $[0, \infty)$, donde 0 indica una separación mínima entre grupos y valores mayores representan una mayor separación entre ellos. Dunn se calcula de la siguiente forma (Ecuación 2.37):

$$D = min_{1 \leqslant i \leqslant k} \left(min_{i+1 \leqslant j \leqslant k} \left(\frac{dist(C_i, C_j)}{max_{1 \leqslant l \leqslant k} diam(C_k)} \right) \right) \quad (2.37)$$

  donde $dist(C_i, C_j)$ es la distancia entre grupos C_i y C_j y $diam(C_k)$ es el diámetro del grupo C_k.

- Coeficiente de silueta [Starczewski et al., 2015]: mide el grado de compactación y separación de los datos en los grupos. En concreto, la compactación cuantifica la distancia de un dato con otros datos del mismo grupo, mientras que la separación mide la distancia entre un dato y los datos de otros grupos. El coeficiente de silueta de un dato denominado $S(i)$, puede calcularse mediante la siguiente (Ecuación 2.38):

$$S(i) = \frac{(b(i) - a(i))}{max\{a(i), b(i)\}} \quad (2.38)$$

donde $a(i)$ representa la distancia media entre el dato i y todos los demás datos del mismo grupo y $b(i)$ representa la distancia media entre el dato i y todos los datos de los grupos vecinos más cercanos. El resultado del cálculo del coeficiente de silueta debe estar en el rango $[-1, 1]$, donde -1 indica una mala asignación de los datos a los grupos, 0 indica que los grupos se superponen y valores cercanos a 1 indican una buena separación de los grupos.

- Índice de Calinski-Harabasz (*Calinski-Harabasz Index* en inglés): también conocido como criterio de la relación de varianza, se define como la relación entre la dispersión total (mide el grado de separación de los grupos) entre grupos y la dispersión interna (mide el grado de compactación de los datos en cada grupo) de los grupos con el resto de los grupos. Cuanto mayor sea el valor de la (Ecuación 2.39), mejores serán los resultados.

$$CH(b, Sw) = \frac{trace\,(S_b)}{trace\,(S_w)} . \frac{n_p - 1}{n_p - k} \tag{2.39}$$

donde S_b es la matriz de dispersión entre grupos, S_w la matriz de dispersión interna, k es el número de grupos y n_p es el número de muestras agrupadas. El valor resultante de calcular Índice de Calinski-Harabasz se encuentra en el rango $[0, \infty)$, donde los valores más altos indican grupos más compactos y bien separados, lo cual sugiere una mejor calidad de agrupamiento.

- Índice de Davies-Bouldin (*Davies-Bouldin Index* en inglés): es una medida de calidad encargada de medir la compactación y separación de los grupos. Índice de Davies-Bouldin se calcula (Ecuación 2.40) de la siguiente forma:

$$BD(i, j) = \frac{1}{c} \sum_{i=1}^{c} Max_{i \neq j} \left\{ \frac{d\,(X_i) + d\,(X_j)}{d\,(c_i, c_j)} \right\} \tag{2.40}$$

donde c indica el número de grupos, i,j son las clases de los grupos, $d(X_i)$ y $d(X_j)$ son todas las muestras en los grupos i y j a sus respectivos centroides de los grupos y $d\,(c_i, c_j)$ es la distancia entre los centroides. El valor de calcular el índice de Davies-Bouldin debe estar en el rango $[0, \infty)$, donde los valores más próximos a 0 indican un alto grado de compactación y separación.

- Índice de Bic (*Bic Index* en inglés): su objetivo principal es seleccionar el modelo que mejor se ajuste a los datos, teniendo en cuenta tanto la capacidad de ajuste del modelo como su complejidad. Índice de Bic penaliza la complejidad del modelo, lo que significa que modelos más simples tienen una ventaja sobre modelos más complejos si ambos ajustan los datos de manera similar (Ecuación 2.41):

$$BIC = -\ln(L) + \nu \ln(n) \tag{2.41}$$

  donde n es el número de datos, L es la verosimilitud de los parámetros para generar los datos en el modelo, y ν es el número de parámetros libres en el modelo gaussiano. El rango de valores que puede tomar índice BIC se encuentran en el rango $(-\infty, \infty)$. Para los casos donde los valores son menores indican un mejor modelo de agrupamiento.

Como resumen del conjunto de medidas internas mencionadas anteriormente se establece una agrupación de las medidas en base a la compactación y separación. La tabla 2.1 refleja esta comparativa.

Medidas	Compactación	Separación
Conectividad	X	
Dunn	X	X
Índice de Bic		X
Índice de Calinski-Harabasz	X	X
Índice de Davies-Bouldin	X	X
Coeficiente de silueta	X	X

Tabla 2.1: Agrupación de las medidas internas en compactación y separación.

2.6. Aplicaciones en ámbitos reales

El interés en esta técnica ha aumentado con el tiempo, surgiendo nuevas líneas de investigación emergentes de algoritmos de agrupamiento automático [Ezugwu, 2020]. Debido a su gran versatilidad, los algoritmos de agrupamiento se pueden aplicar en una amplia gama de campos, como se detalla a continuación:

- Energía: el sector energético ha logrado éxitos significativos en la implementación de algoritmos de agrupamiento. [Agbehadji et al., 2021] presenta un enfoque innovador para optimizar el uso de energía en sensores, inspirado en el comportamiento animal. Su objetivo es prolongar la vida útil de los sensores IoT en una red de procesamiento y transmisión continua de paquetes, lo que hace que la optimización del ahorro energético sea fundamental. Para lograrlo, compara los algoritmos KSA-DEEC y WSAMP-DEEC en modo de agrupamiento. La caída de los costos de energía eólica y solar, ha adquirido una importancia creciente. [Tanoto et al., 2020] propone un modelo híbrido que combina el algoritmo de agrupamiento k-means y un mapa autoorganizado basado en redes neuronales. Su objetivo es integrar tecnologías con patrones similares para planificar una industria eléctrica equilibrada en términos de costos, seguridad e impacto ambiental. Además, los algoritmos de agrupamiento se utilizan para crear perfiles de consumidores a partir de datos de contadores inteligentes de manera rápida y eficiente [Shamim et al., 2020]. La recopilación de datos de consumo de energía se ha vuelto más fácil gracias a la disminución del coste de los sensores inteligentes. Esta recopilación de datos permite diseñar sistemas de cogeneración que puedan adaptarse a los perfiles de demanda energética de forma más eficiente, eligiendo el tipo correcto de tecnología de cogeneración, la estrategia de operación y, si son necesarios, el tamaño de los almacenamientos de energía. [Vialetto et al., 2020] utiliza el algoritmo de partición k-means junto con la medida de calidad el coeficiente de silueta [Starczewski et al., 2015] para determinar el número óptimo de grupos para agrupar los datos de demanda de energía.

- Fabricación: es otro de los sectores que se beneficia del uso de métodos de agrupamiento, similares a los mencionados en el sector del transporte. Principalmente, se utilizan métodos híbridos que combinan enfoques analíticos con métodos de agrupamiento. [Delgoshaei et al., 2019] revisó los métodos híbridos de agrupamiento y los algoritmos de búsqueda, como las metaheurísticas, en el diseño de sistemas de fabricación de dispositivos. Asimismo, se han propuesto mejoras en el algoritmo k-means con el fin de determinar un número óptimo de grupos en la industria inteligente [Yin, 2020]. Otro caso de éxito radica en su aplicación en la fabricación para agrupar proveedores según sus necesidades específicas [Sabbagh et al.,

2020]. [Subramaniyan et al., 2020] propuso la creación de agrupamientos temporales de datos mediante algoritmos jerárquicos aglomerativos para detectar cuellos de botella en los sistemas de fabricación.

- Financiero: se caracteriza por la seguridad de sus sistemas. Los sistemas requieren un tratamiento inteligente de los datos por parte de los sistemas automatizados. [Boyko et al., 2019] propone la combinación de varios algoritmos, incluido el k-means, para desarrollar una plantilla que permita analizar el comportamiento del usuario en función de su ubicación prevista para el próximo mes. Además, es uno de los sectores líderes en la adopción de la digitalización. Sin embargo, su progreso a veces se ve obstaculizado por amenazas, como el lavado de dinero, que puede abordarse mediante técnicas de agrupamiento espacial basadas en densidad, como Dbscan. Un ejemplo de aplicación de este algoritmo es en la detección y notificación de transacciones bancarias sospechosas [Yang Yang et al., 2014]. La ubicación de los cajeros automáticos y la incertidumbre en los datos demográficos de la población son elementos esenciales para adaptar los servicios financieros a las necesidades de los clientes. Por lo tanto, [Kisore et al., 2017] propone una modificación del algoritmo de agrupación basado en la densidad generalizada (Gdbscan) para agrupar los cajeros en función de parámetros socioeconómicos.

- Sanidad: es un sector en constante evolución y con una gran cantidad de datos provenientes de diversas fuentes [J. Zhang et al., 2022], [Shafqat et al., 2020]. La COVID-19 ha provocado que la atención sanitaria de todos los países del mundo se haya visto colapsada. La forma más efectiva de mejorar la atención sanitaria es analizando los sistemas que han demostrado ser eficientes en la prestación de servicios sanitarios. En este contexto, [Zubair et al., 2021] propone un método basado en k-means para determinar los centroides iniciales de los grupos, lo que permite mejorar el análisis de la calidad sanitaria de los países de una manera más rápida y con menos iteraciones. La detección de valores atípicos es un área de investigación crucial en el aprendizaje automático y la Ciencia de Datos. Los valores atípicos en un conjunto de datos pueden limitar significativamente su utilidad en aplicaciones del mundo real. [Ray et al., 2022] propone una técnica que permite detectar valores atípicos combinando el conocimiento de algoritmos como k-means, k-means++ y fuzzy c-means. Utiliza una técnica probabilística novedosa para eliminar los valores atípicos y lograr índices de calidad significativamente altos. Otro aspecto importante

es la identificación de patrones en los procesos clínicos y sanitarios para estudiar la progresión de los pacientes. Para abordar este desafío, [Johns et al., 2020] propone una medida de distancia que puede combinarse con algoritmos de agrupamiento tradicionales para identificar patrones en la evolución de los pacientes.

- Transporte: un sector tan destacado como es del transporte y la logística se beneficia de las técnicas de agrupamiento al ofrecer alternativas rápidas y efectivas para mitigar la congestión del tráfico en áreas específicas. Esto se logra mediante una búsqueda espacial que permite a los responsables tomar decisiones para mejorar las operaciones diarias en zonas con congestión frecuente [Almannaa et al., 2019]. Estas técnicas se aplican en diversos tipos de transporte, incluidos el transporte de mercancías peligrosas [De Luca et al., 2011], el transporte por carretera [J. Lu et al., 2013] y el transporte urbano/público [Sfyridis et al., 2020]. Además, se utilizan algoritmos de agrupamiento en combinación con algoritmos evolutivos para identificar y gestionar situaciones de alto riesgo en buques, teniendo en cuenta tanto la carga como los efectos ambientales relacionados [Tran, 2020].

2.7. Librería de *Clustering* en R

Existe un gran número de librerías desarrolladas para problemas de agrupamiento. Dentro de ellas, destacan las desarrolladas en R por su versatilidad y amplia difusión entre la comunidad investigadora. En el problema de agrupamiento, existen librerías en R que implementan algoritmos, medidas de calidad, distancias y funcionalidades para la comparación de algoritmos. Sin embargo, las librerías existentes presentan algunos problemas:

- No permiten trabajar con diferentes formatos.
- Los algoritmos se centran en distribuir los datos en grupos, pero no en crear grupos de calidad.
- Permiten trabajar con sólo un conjunto de datos, impidiendo comparar diferentes conjuntos de datos de manera simultánea.
- No es posible evaluar los grupos creados aplicando un conjunto de medidas de calidad de forma concurrente. Para el caso de las medidas externas al no disponer de datos etiquetados, las medidas son evaluadas de forma

individual para cada una de las variables del conjunto de datos. Este proceso lo deben realizar los propios usuarios de las librerías, lo que implica mucho tiempo y un esfuerzo considerable.

- No es posible evaluar los resultados para un rango de grupos.
- En la actualidad existen pocas librerías que dispongan de una interfaz de usuario que facilite su uso.

Con el propósito de resolver los problemas mencionados de las librerías de agrupamiento en R, se ha desarrollado una propuesta, la librería *Clustering* [Pérez-Martos et al., 2023], [Pérez-Martos et al., 2022], que: permite evaluar y medir la calidad de los grupos creados; dispone de una interfaz de usuario fácil de usar y muy útil para configurar y ejecutar experimentos de forma rápida y sencilla sin necesidad de conocer los parámetros de los algoritmos, facilitando y agilizando el análisis y la comparación de los resultados de diferentes algoritmos. De esta forma, las ventajas de la librería *Clustering* respecto a otras librerías R existentes se pueden resumir como:

- Puede trabajar tanto con un conjunto de datos como con un directorio que contenga varios conjuntos de datos.
- Permite realizar un estudio experimental con diferentes algoritmos, medidas de distancia, número de grupos, y medidas de calidad tanto interna como externa. En el análisis de las medidas de calidad externas, se evalúan todas las variables del conjunto de datos, con el objetivo de seleccionar la variable que mejor comportamiento tiene.
- Dispone de una completa e intuitiva interfaz de usuario. A día de hoy, solo seis librerías en R para agrupamiento disponen de interfaz de usuario: ProjectionBasedClustering, OpenRepGrid, dtwclust, visxhclust, rainette y VarSelLCM. Indicar que la interfaz VarSelLCM, OpenRepGrid, dtwclust y visxhclust no funcionan mientras que ProjectionBasedClustering y rainette no permiten comparar algoritmos ni medidas de calidad.

La librería *Clustering* incorpora los algoritmos más relevantes de la *Task View*[1] de grupos, especialmente de las secciones de algoritmos jerárquicos y particionales, utilizando para ello las implementaciones originales de sus autores. La

[1] https://cran.r-project.org/web/views/Cluster.html

tabla 2.2 muestra un estudio comparativo de las librerías de agrupamiento descargadas con más frecuencia y la librería *Clustering*, cuyas columnas recogen la siguiente información:

- Medidas internas y externas: indica si la librería incorpora medidas para validar la calidad del agrupamiento. Para los casos en los que no se incorpore ninguna medida se indica no. Para el resto de los casos se especifica qué medida o medidas se implementan.
- Validación externa del conjunto de datos: manifiesta si las librerías hacen una evaluación externa de todos los atributos del conjunto de datos. En el caso de que no implemente esta funcionalidad se indica no.
- Comparativa por algoritmo: refleja si es posible comparar múltiples algoritmos de agrupamiento de forma automática, en lugar de hacer ejecuciones secuenciales de cada algoritmo para comparar posteriormente los resultados. Esta utilidad es bastante interesante, si deseamos ver el comportamiento de diferentes algoritmos y/o variar los parámetros de entrada.
- Comparativa por métricas: detalla si es posible analizar los resultados de los algoritmos por: medidas de distancia, calidad o número de grupos.
- Interfaz de usuario: indica qué librerías implementan una interfaz de usuario.

Librerías	Algoritmos	Medidas interna y externa	Validación externa del conjunto de datos	Comparativa por algoritmo	Comparativa por métricas	Interfaz de usuario
pvclust	Pvclust y pvpick	No	No	No	No	No
cluster	Agnes, clara, diana, fanny, mona y pam	Coeficiente de silueta	No	No	No	No
apcluster	Aggexcluster, apclusterk y apclusterl	No	No	No	No	No
ClusterR	Ap_affinity_propagation, gmm, kmeans_arma, kmeans_rcpp y mininBatchKmeans	Coeficiente de Mirkin, entropía, índice ajustado de Rand, índice de Fowlkes-Mallows, índice de Rand y pureza	No	No	No	No
amap	Hcluster y k-means	No	No	No	No	No
Clustering	Aggexcluster, agnes, apclusterk, clara, daisy, diana, fanny, gmm, hcluster, kmeans_arma, kmeans_rcpp, minibatchkmeans, mona, pam, pvclust y pvpick	Conectividad, dunn, entropía, índice de Fowlkes-Mallows, precisión, recuperación, coeficiente de silueta, valor-F y variación de la información	Si	Algoritmo, grupos, medida de distancia, medidas externa e interna	Si	Si

Tabla 2.2: Comparativa de las funcionalidades implementadas por las librerías jerárquicas y particionales más destacadas de la Task View [a] de grupos en R.

[a] https://cran.r-project.org/web/views/Cluster.html

A continuación, se detalla la arquitectura y funcionalidades de la librería *Clustering*. Se cierra esta sección con un caso práctico de uso de la librería.

2.7.1. Arquitectura

La creación de la librería *Clustering* aporta mejoras significativas en el análisis del agrupamiento al ofrecer una serie de ventajas distintivas como las ya indicadas: posibilidad de analizar uno o varios conjunto de datos, puede trabajar con diversos algoritmos, permite la escalabilidad mediante la incorporación de nuevos algoritmos, admite la utilización de múltiples medidas de distancia durante la ejecución, opera sobre un rango de grupos, incorpora métricas de calidad para evaluar la compactación y separación de los grupos además de proporcionar una interfaz de usuario intuitiva que simplifica el uso de esta librería.

Para ofrecer todas estas características la librería está compuesta por los siguientes elementos:

- 5 librerías: amap [Team, 2024], apcluster [Bodenhofer et al., 2011], cluster [Maechler et al., 2019], ClusterR [Struyf et al., 1997] y pvclust [Suzuki et al., 2006].
- 16 algoritmos: aggExCluster [Bodenhofer et al., 2011], agnes [Lance et al., 1966], apcluster [Bodenhofer et al., 2011], clara [Ramprasanth et al., 2019], daisy [Kaufman et al., 1990], diana [Patnaik et al., 2016], fanny [Kaufman et al., 1990], gmm [Struyf et al., 1997], hcluster [Team, 2024], kmeans_arma [Struyf et al., 1997], kmeans_rcpp [Struyf et al., 1997], miniBatchKmeans [Struyf et al., 1997], mona [Kaufman, 1990], pam [Kaufman, 1990], pvpick [Suzuki et al., 2006] y pvclust [Suzuki et al., 2006].
- 6 medidas externas: entropía [Sripada et al., 2011], variación de la información [Meilă, 2003], precisión [Hanczar et al., 2013], recuperación [Rezaei et al., 2016], valor-f [J. Wu et al., 2009], y índice de Fowlkes-Mallows [Nemec et al., 1988].
- 3 medidas internas: coeficiente de silueta [Starczewski et al., 2015], dunn [Rendón et al., 2011], y conectividad [Saha et al., 2009].

Además de las librerías mencionadas anteriormente, la librería *Clustering* usa otras librerías que son necesarias para la ejecución y visualización de los resultados. El conjunto de dependencias de las que está compuesta la librería se puede visualizar en la figura 2.10, donde los cuadrados grises denotan todas las

importaciones de la librería *Clustering*, mientras que el cilindro rojo indica la dependencia con el core de R. El rectángulo naranja representa la propia librería *Clustering*.

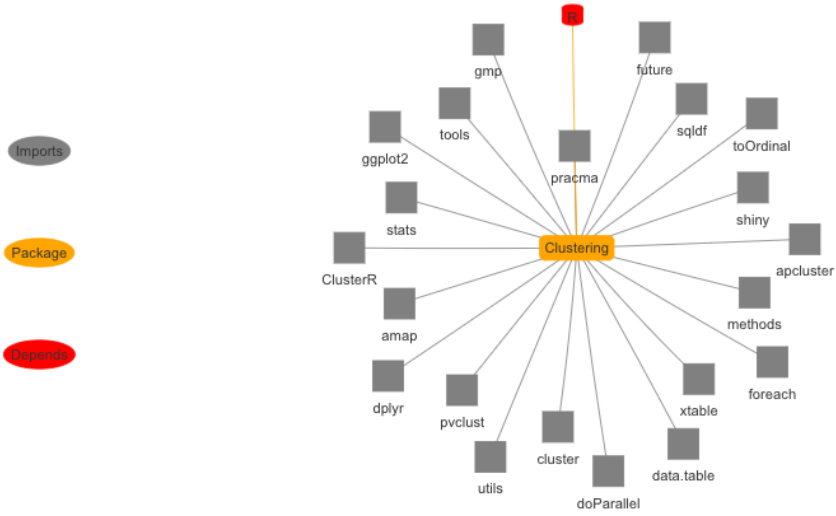


Figura 2.10: Arquitectura de la librería *Clustering*. *Fuente: elaboración propia.*

Los algoritmos que componen la librería son detallados en la tabla 2.3.

Librería	Algoritmo	Distancia	Descripción
amap	hcluster [Team, 2024]	Euclídea	Agrupación jerárquica basada en k-means
apcluster	aggExCluster [Bodenhofer et al., 2011]	Euclídea	Agrupación aglomerativa basada en ejemplos
apcluster	apclusterK [Bodenhofer et al., 2011]	Euclídea, Manhattan y Minkowski	Propagación de afinidades para un número predefinido de grupos
cluster	agnes [Lance et al., 1966]	Euclídea y Manhattan	Anidamiento aglomerativo
cluster	clara [Ramprasanth et al., 2019]	Euclídea y Manhattan	Agrupación de grandes conjuntos de datos
cluster	daisy [KaufmanandP et al., 1990]	Euclídea, Manhattan y Gower	Algoritmo que maneja de manera eficiente tipos de variables, por ejemplo, nominales, ordinales, binarias asimétricas
cluster	diana [Patnaik et al., 2016]	Euclídea	Análisis divisivo de agrupamiento
cluster	fanny [Kaufman et al., 1990]	Euclídea y Manhattan	Agrupamiento difuso
cluster	mona [Kaufman, 1990]	-	Análisis monotético para agrupar variables binarias
cluster	pam [Kaufman, 1990]	Euclídea y Manhattan	Particionamiento basado en medoides
ClusterR	gmm [Struyf et al., 1997]	Euclídea y Manhattan	Agrupación de modelos de mezclas gaussianas
ClusterR	kmeans_arma [Struyf et al., 1997]	-	K-means usando la librería Armadillo
ClusterR	kmeans_rcpp [Struyf et al., 1997]	-	K-means usando RcppArmadillo
ClusterR	miniBatchKmeans [Struyf et al., 1997]	-	MiniBatchKmeans usando RcppArmadillo
pvclust	pvclust [Suzuki et al., 2006]	Correlación de Pearson	Realiza un bootstrap multiescala
pvclust	pvpick [Suzuki et al., 2006]	Euclídea y Correlación de Pearson	Realiza un bootstrap multiescala

Tabla 2.3: Algoritmos de agrupamiento integrados en la librería *Clustering*.

Los algoritmos integrados en la librería *Clustering* pueden ser usados a través del método *Clustering::clustering()*, que constituye el método principal de esta librería. Este método se encarga de diversos aspectos como son:

- Ajustar correctamente los parámetros de cada algoritmo.
- Ejecutar en paralelo los algoritmos seleccionados y recopilar sus resultados para cada conjunto de datos.
- Evaluar la calidad de los grupos obtenidos.
- Presentar y facilitar la gestión de toda esta información de manera intuitiva.

Al ejecutar el método *Clustering::clustering()*, se devuelve un objeto llamado *clustering*. Este objeto contiene información sobre qué algoritmos se han ejecutado, las métricas empleadas, indicadores sobre medidas internas o externas, y los resultados. La librería ofrece múltiples funciones auxiliares para analizar y evaluar los resultados obtenidos. La librería puede ser descargada desde el directorio de CRAN [2].

2.7.2. Funcionalidades

La librería *Clustering* incluye las siguientes funcionalidades:

- *Clustering::clustering():* es la función principal de la librería. El resultado de ejecutar este método es un objeto de tipo *clustering*. También exporta los métodos S3 *print()* y *summary()*. El método *summary()* muestra la estructura de datos sin codificación y un resumen con la información básica sobre el conjunto de datos utilizado. También es posible la ordenación y filtrado de los resultados obtenidos. El operador "["filtra la información haciendo uso de la librería dplyr.
- *Medidas externas:* se encargan de asegurar la calidad de los grupos usando los atributos del conjunto como clases. Los métodos implementados por la librería para validar las medidas externas son:

[2] https://cran.r-project.org/web/packages/Clustering/index.html

  - *Clustering::best_ranked_external_metrics():* el resultado de ejecutar este método es una tabla con los atributos del conjunto de datos con mejor comportamiento por algoritmo, medida de distancia y número de grupos agrupados en ranking por los mejores valores.
  - *Clustering::evaluate_best_validation_external_by_metrics():* este método agrupa los datos por algoritmo y medidas de distancia, en lugar de obtener los mejores atributos del conjunto de datos.
  - *Clustering::evaluate_validation_external_by_metrics():* agrupa los resultados de la ejecución por algoritmo.
  - *Clustering::result_external_algorithm_by_metrics():* obtiene los resultados para un algoritmo dado agrupado por número de grupos.

- *Medidas internas:* incluye el mismo conjunto de métodos que los mencionados para las medidas externas, pero cambiando externa por interna en el nombre del método:

  - *Clustering::best_ranked_internal_metrics():* la ejecución de este método obtiene los mejores atributos agrupados por algoritmo, medida de distancia y número de grupos.
  - *Clustering::evaluate_best_validation_internal_by_metrics():* este método agrupa los datos por algoritmo y medidas de distancia, en lugar de agruparlo por atributos.
  - *Clustering::evaluate_validation_internal_by_metrics():* agrupa los resultados de la ejecución por algoritmo.
  - *Clustering::result_internal_algorithm_by_metrics():* obtiene los resultados de un algoritmo dado agrupado por número de grupos.

- *Clustering::plot_clustering()*: este método visualiza los resultados del agrupamiento mediante un gráfico de barras. El gráfico representa la distribución del algoritmo por número de particiones y métrica de calidad que puede ser externa o interna.

- *Clustering::export_external_file()*: exporta los resultados de las medidas externas a formato LaTeX

- *Clustering::export_internal_file()*: tiene la misma funcionalidad que el método anterior, pero en este caso exporta las medidas internas a un fichero en formato LaTeX

2.7.3. Caso práctico

Una vez presentada la arquitectura y los métodos que ofrece la librería *Clustering*, procederemos a detallar cómo se puede utilizar la librería tanto desde la línea de comandos como desde la interfaz de usuario.

- **Uso mediante la Interfaz de usuario:** para ejecutar la librería desde la línea de comandos, debemos tener instalado R en nuestro sistema. Una vez comprobado que la instalación es correcta, abrimos un terminal y ejecutamos los siguientes comandos tal y como se indica en la figura 2.11:

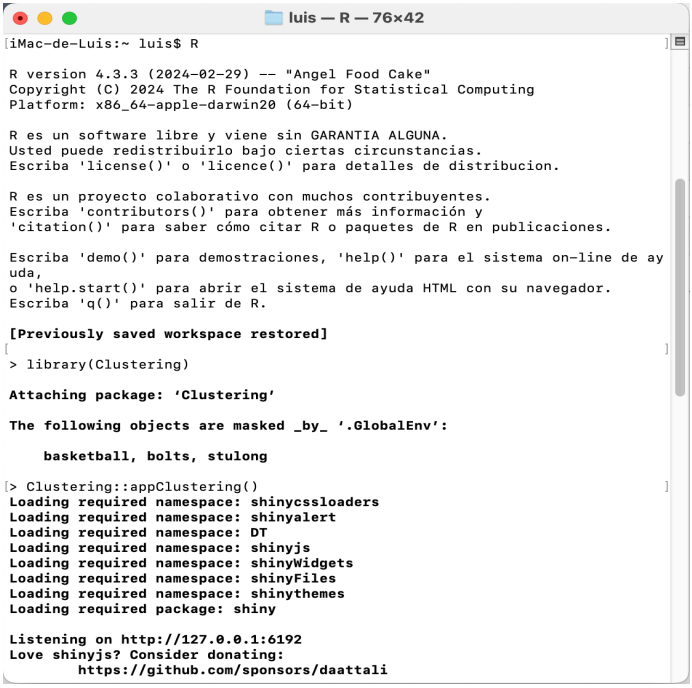


Figura 2.11: Ejemplo de cómo ejecutar la interfaz de usuario desde la consola del sistema. *Fuente: elaboración propia.*

En la figura 2.11 se carga la librería *Clustering* y ejecutamos un método de la librería llamado *Clustering::appClustering()*. Tras la ejecución de estas instrucciones aparece la interfaz gráfica, tal como se muestra en las figuras 2.12 y 2.13.

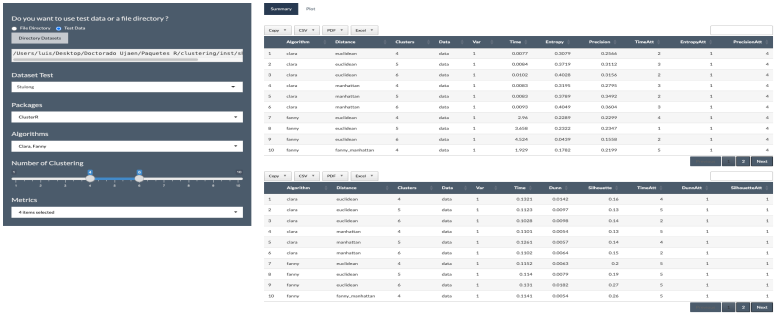


Figura 2.12: Interfaz de usuario de la librería *Clustering*. Análisis de datos por algoritmos, medidas de calidad y número de grupos. *Fuente: elaboración propia.*

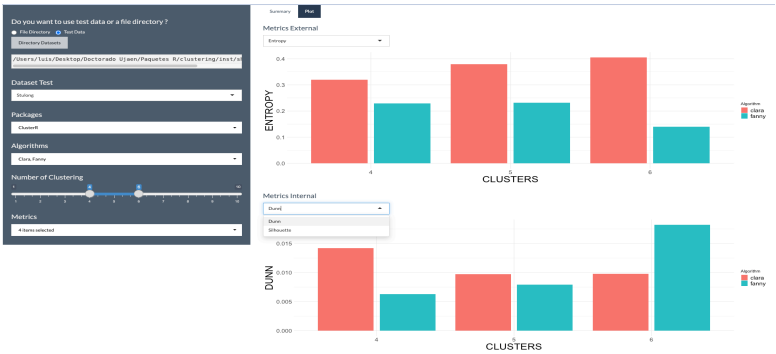


Figura 2.13: Representación gráfica de las medidas de calidad externa e interna por algoritmo y número de grupos usando la interfaz web de la librería *Clustering*. *Fuente: elaboración propia.*

La figura 2.12 está divida en dos partes. En la parte izquierda, se presentan los parámetros con los que trabaja la librería *Clustering*: conjuntos de datos (que pueden ser predefinidos o configurados de forma manual in-

dicando la ruta del directorio donde se ubica el listado de conjuntos de datos), algoritmos a ejecutar (se puede ejecutar de forma individual, conjunta o seleccionando de forma individual los algoritmos de las librerías incluidas como dependencias), el número de grupos y las medidas de calidad. En la parte central se cargan de forma dinámica los resultados a partir de los criterios indicados. Los resultados se desglosan en dos tablas. En la parte superior aparecen los resultados de evaluar las medidas externas y en la parte inferior las medidas internas. No siempre van a aparecer ambas tablas, ya que la aparición de cada una de ellas va a depender de los parámetros de entrada.

Si por el contrario queremos ver como se representa la distribución de los datos de forma gráfica, debemos hacer click sobre la pestaña de la parte superior denominada ***Plot***.

En la figura 2.13 tenemos una representación de la distribución de las medidas de calidad por algoritmo y número de grupos. Además, tenemos la posibilidad de ir jugando con las medidas y ver la distribución de los datos para cada una de ellas.

- **Uso desde la terminal del sistema:** para el caso de querer ejecutar la librería lo hacemos desde la línea de comandos. La librería se compone de un conjunto de funcionalidades agrupadas en métodos. Para utilizar los métodos de la librería *Clustering* en R, es fundamental ejecutar el método principal llamado *Clustering::clustering()* con una serie de parámetros tal y como se mencionó en la subsección 2.7.2. La función *Clustering::clustering()* es la que permite analizar el agrupamiento utilizando diferentes algoritmos además de comparar sus resultados para determinar que algoritmo se comporta mejor con los datos proporcionados. Para ilustrar el funcionamiento del método *Clustering::clustering()* en R, consideremos un caso práctico utilizando el conjunto de datos *Clustering::basketball* integrado en la librería. El conjunto de datos está compuesto de cinco atributos, como se describe a continuación:

  - assists_per_minuteReal
  - heightInteger
  - time_playedReal
  - ageInteger
  - points_per_minuteReal

Cuando veamos tablas cuyas columnas tienen nombres con el sufijo Att, significa que los valores de la columna hacen referencia a los atributos del conjunto de datos que se está evaluando. Es decir, si aparece el valor 5th se refiere al quinto atributo del conjunto de datos. Para el caso de *Clustering::basketball* se refiere al atributo *points_per_minuteReal*.

Vamos a ejecutar la función *Clustering::clustering()* para agrupar los datos en un número de grupos que varía entre 3 y 4, utilizando el algoritmo *pam* y evaluando las métricas *Precision* y *Dunn*. Un ejemplo de ejecución del método se puede visualizar en el código 2.1.

```
# Cargamos la librería
library(Clustering)

# Ejecutamos el método.
resultado <- Clustering::clustering(df = Clustering::basketball, min =
 3, max = 4, algorithm = 'pam', metrics = c('Precision','Dunn'))
```

Código 2.1: Ejemplo de ejecución del método Clustering::clustering().

Nota: la carga de la librería solo es necesario realizarlo una sola vez. El código 2.1, muestra la ejecución el método *Clustering::clustering()* para un conjunto de datos.

El resultado de la ejecución es un objeto *clustering* con la estructura descrita en el código 2.2:

```
summary(resultado)
Object of class 'clustering'Result:

Algorithm   Distance   Clusters    Data    Var  Time  Precision
   pam      euclidean     3     basketball 1st 0.0061  0.1739
   pam      euclidean     3     basketball 2nd 0.0071  0.0813
   pam      euclidean     3     basketball 3rd 0.0117  0.0006
   pam      euclidean     3     basketball 4th 0.0132  0.0000
   pam      euclidean     3     basketball 5th 0.0133  0.0000
   pam      euclidean     4     basketball 1st 0.0043  0.1673
   pam      euclidean     4     basketball 2nd 0.0069  0.1015
   pam      euclidean     4     basketball 3rd 0.0076  0.0000
   pam      euclidean     4     basketball 4th 0.0142  0.0000
   pam      euclidean     4     basketball 5th 0.0243  0.0000
   pam      manhattan     3     basketball 1st 0.0037  0.1851
   pam      manhattan     3     basketball 2nd 0.0038  0.0775
   pam      manhattan     3     basketball 3rd 0.0063  0.0006
   pam      manhattan     3     basketball 4th 0.0064  0.0000
   pam      manhattan     3     basketball 5th 0.0066  0.0000
   pam      manhattan     4     basketball 1st 0.0041  0.1718
   pam      manhattan     4     basketball 2nd 0.0043  0.0952
   pam      manhattan     4     basketball 3rd 0.0064  0.0009
   pam      manhattan     4     basketball 4th 0.0067  0.0000
   pam      manhattan     4     basketball 5th 0.0069  0.0000

Internal Metrics:
[1] TRUE

External Metrics:
[1] TRUE

Number of Algorithms:
[1] 1

Number of Measures:
[1] 2

Total elements:
[1] 20

Mean time for evaluation of external metrics:
[1] "0.0082"

Metric mean Precision:
[1] "0.0528"
```

Código 2.2: Atributos que componen el objeto *clustering*.

En el código 2.2 tenemos una tabla con las columnas: `Algorithm` indica el nombre del algoritmo, `Distance` representa la medida de distancia empleada, `Clusters` es el número de grupos, y `Data` es el nombre del conjunto de datos analizado. La librería *Clustering* intenta encontrar el atributo que proporciona la mejor partición de los datos sobre las métricas externas utilizadas. Es obligatorio indicar al menos una medida externa en el método *Clutsering::clustering()*.

Para poder ejecutar el resto de los métodos existentes en la librería es necesario pasar como parámetro el objeto *clustering*, además de otros parámetros que varían según el método. Para analizar los resultados de las medidas externas debemos de seguir el siguiente proceso.

- En primer lugar, ejecutamos el método *Clustering::best_ranked_external_metrics()* para obtener los mejores resultados para las medidas externas agrupados por algoritmo, medida de distancia y número de grupos.

```
Clustering::best_ranked_external_metrics(resultado)

Algorithm  Distance   Clusters  Var  Time  Precision TimeAtt
  PrecisionAtt
   pam     euclidean     3      1st 0.0071  0.1739     5th      2nd
   pam     euclidean     4      1st 0.0056  0.1673     5th      2nd
   pam     manhattan     3      1st 0.0066  0.1851     5th      2nd
   pam     manhattan     4      1st 0.0077  0.1718     4th      2nd
```

Código 2.3: Ejemplo de ejecución del método Clustering::best_ranked_external_-metrics() a partir del objeto *clustering*.

El resultado obtenido tras la ejecución del código 2.3 es una tabla con las medidas externas agrupadas por algoritmo, medida de distancia y número de grupos. La columna `Var` refleja qué atributo del conjunto de datos se ha utilizado como objetivo. Las columnas `Time` y `Precision`, muestran el valor de la métrica en relación con el atributo `Var` utilizado como objetivo, así como el tiempo empleado en calcular dicho valor.

Es importante destacar que las columnas terminadas en `Att`, muestran el atributo del conjunto de datos con mayor influencia en las métricas analizadas.

- Para aquellos casos en los que es necesario filtrar los resultados por una medida concreta podemos hacerlo con el método *Clustering::evaluate_best_validation_external_by_metrics()*. Este método recibido como parámetro el objeto *clustering* y la medida externa que deseamos filtrar. No podemos aplicar cualquier medida externa, solo podemos seleccionar una medida de entre las disponible en el objeto *clustering*. Continuando con la exposición sobre la forma de ejecutar los métodos, se presenta el método *Clustering::evaluate_best_validation_external_by_metrics()* en el código 2.4.

```
Clustering::evaluate_best_validation_external_by_metrics(resultado
 , 'Precision')

Algorithm  Distance   Clusters  Time    Precision TimeAtt
 PrecisionAtt
   pam      euclidean     3      0.0071   0.1739     5th       2nd
   pam      manhattan     3      0.0066   0.1851     5th       2nd
```

Código 2.4: Ejemplo de filtrado de los resultados por la medida *Precision*.

Podemos apreciar que al obtener los mejores resultados para la medida externa *Precision*, el sistema se ha quedado con los resultados obtenidos para 3 grupos, ya que los valores que están más próximos a 1, son considerados por el sistema más relevantes para la métrica indicada.

- Para obtener el mejor resultado por algoritmo, medida de distancia y número de grupos para el conjunto de datos debemos usar el método *Clustering::evaluate_validation_external_by_metrics()*.

```
Clustering::evaluate_validation_external_by_metrics(resultado)

Algorithm  Time    Precision TimeAtt PrecisionAtt
   pam     0.0077   0.1851     5th       2nd
```

Código 2.5: Resultados obtenidos tras la ejecución del método Clustering::evaluate_validation_external_by_metrics().

El resultado del código 2.5 destaca que el atributo con el que se consigue un agrupamiento más relevante para el conjunto de datos es *heightInteger*. Si además del algoritmo, queremos determinar la medida de distancia podemos hacerlo haciendo uso del método indicado en el código 2.6.

```
Clustering::result_external_algorithm_by_metric(resultado,'
 Precision')

Algorithm  Distance   Clusters  Time  Precision TimeAtt
 PrecisionAtt
    pam     manhattan     3     0.0066   0.1851    5th       2nd
```

Código 2.6: Filtrado de los resultados con mejor relevancia por algoritmo y medida de distancia.

- Para poder generar un gráfico con los resultados externos por medidas podemos hacerlo a través del método *Clustering::plot_clustering()*. Este método al igual que los anteriores recibe como parámetro el objeto *clustering* y la medida de calidad. La forma de hacerlo es mediante la ejecución del código 2.7, y el resultado queda reflejado en la figura 2.14.

```
Clustering::plot_clustering(resultado,'Precision')
```

Código 2.7: Método para representar gráficamente el resultado de *Precision* por algoritmo y número de grupos.

- Para finalizar, podemos ordenar las columnas en orden ascendente y descendente, exportar los resultados a LaTeX y filtrar las columnas por una serie de valores tal y como queda reflejado en el código 2.8 y 2.9.

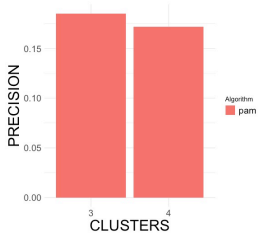


Figura 2.14: Visualización gráfica del resultado para la medida *Precision*. *Fuente: elaboración propia.*

```
# Para ordenar los campos en orden ascendente cambiar T por F.
sort(resultado, T, 'Precision')

Algorithm  Distance   Clusters    Data      Var   Time   Precision
   pam     manhattan    3       basketball 1st  0.0066   0.1851
   pam     euclidean    3       basketball 1st  0.0071   0.1739
   pam     manhattan    4       basketball 1st  0.0077   0.1718
   pam     euclidean    4       basketball 1st  0.0056   0.1673
   pam     euclidean    4       basketball 2nd  0.0091   0.1015
   pam     manhattan    4       basketball 2nd  0.0079   0.0952
   pam     euclidean    3       basketball 2nd  0.0072   0.0813
   pam     manhattan    3       basketball 2nd  0.0074   0.0775
   pam     manhattan    4       basketball 3rd  0.0114   0.0009
   pam     euclidean    3       basketball 3rd  0.0173   0.0006
   pam     manhattan    3       basketball 3rd  0.0098   0.0006
   pam     euclidean    3       basketball 4th  0.0174   0.0000
   pam     euclidean    3       basketball 5th  0.0306   0.0000
   pam     euclidean    4       basketball 3rd  0.0117   0.0000
   pam     euclidean    4       basketball 4th  0.0137   0.0000
   pam     euclidean    4       basketball 5th  0.0151   0.0000
   pam     manhattan    3       basketball 4th  0.0124   0.0000
   pam     manhattan    3       basketball 5th  0.0154   0.0000
   pam     manhattan    4       basketball 4th  0.0151   0.0000
   pam     manhattan    4       basketball 5th  0.0202   0.0000
```

Código 2.8: Ejemplo de ordenación de los resultados por *Precision* en orden descendente.

```
Clustering::export_file_external(resultado, '/Users/luis/Desktop/'
 )

\begin{table}[ht]
\centering
\begin{tabular}{rllllrrr}

    \hline

    Algorithm & Distance & Clusters & Data & Var & Precision &
 TimeAtt & PrecisionAtt \\

    \hline

    pam & euclidean & 3 & basketball & 1 & 0.17 & 5 & 2 \\
    pam & euclidean & 3 & basketball & 2 & 0.08 & 2 & 4 \\
    pam & euclidean & 3 & basketball & 3 & 0.00 & 4 & 3 \\
    pam & euclidean & 3 & basketball & 4 & 0.00 & 1 & 1 \\
    pam & euclidean & 3 & basketball & 5 & 0.00 & 3 & 5 \\
    pam & euclidean & 4 & basketball & 1 & 0.17 & 5 & 2 \\
    pam & euclidean & 4 & basketball & 2 & 0.10 & 1 & 4 \\
    pam & euclidean & 4 & basketball & 3 & 0.00 & 3 & 1 \\
    pam & euclidean & 4 & basketball & 4 & 0.00 & 2 & 3 \\
    pam & euclidean & 4 & basketball & 5 & 0.00 & 4 & 5 \\
    pam & manhattan & 3 & basketball & 1 & 0.19 & 5 & 2 \\
    pam & manhattan & 3 & basketball & 2 & 0.08 & 2 & 4 \\
    pam & manhattan & 3 & basketball & 3 & 0.00 & 3 & 3 \\
    pam & manhattan & 3 & basketball & 4 & 0.00 & 1 & 1 \\
    pam & manhattan & 3 & basketball & 5 & 0.00 & 4 & 5 \\
    pam & manhattan & 4 & basketball & 1 & 0.17 & 4 & 2 \\
    pam & manhattan & 4 & basketball & 2 & 0.10 & 2 & 4 \\
    pam & manhattan & 4 & basketball & 3 & 0.00 & 3 & 3 \\
    pam & manhattan & 4 & basketball & 4 & 0.00 & 1 & 1 \\
    pam & manhattan & 4 & basketball & 5 & 0.00 & 5 & 5 \\
   \hline

\end{tabular}

\end{table}

resultado[Precision > 0.14]

Algorithm  Distance   Clusters     Data    Var  Time    Precision
   pam     euclidean    3       basketball 1st  0.0071    0.1739
   pam     euclidean    4       basketball 1st  0.0056    0.1673
   pam     manhattan    3       basketball 1st  0.0066    0.1851
   pam     manhattan    4       basketball 1st  0.0077    0.1718
```

Código 2.9: Ejemplo de exportación de los resultados de las medidas externas a LaTeX y filtrado de las filas por *Precision*.

Los mismos métodos mencionados para las medidas externas están disponibles para las medidas internas y se ejecutan de la misma forma, tal y como se muestra en el siguiente código 2.10.

```
Clustering::best_ranked_internal_metrics(resultado)

Clustering::evaluate_best_validation_internal_by_metrics(resultado,'Dunn
  ')

Clustering::evaluate_validation_internal_by_metrics(resultado)

Clustering::result_internal_algorithm_by_metric(resultado,'Dunn')

Clustering::plot_clustering(resultado,'Dunn')

Clustering::export_file_internal(resultado, '/Users/luis/Desktop/')
```

Código 2.10: Ejemplo de ejecución de los métodos disponibles para las medidas internas.

Hay que destacar que en el apéndice A.1 se presenta un caso de estudio detallado utilizando la librería *Clustering*. En este caso de estudio se utilizan diferentes conjuntos de datos analizados con algoritmos integrados dentro de la librería. Los resultados serán analizados e interpretados para tomar la mejor decisión.

3 Algoritmos evolutivos para agrupamiento

Este capítulo se centra en presentar los algoritmos evolutivos como una técnica de optimización inspirada en la evolución natural. En el contexto del análisis de datos, los algoritmos evolutivos han demostrado ser herramientas poderosas para resolver una gran variedad de problemas, incluido el agrupamiento. Los algoritmos evolutivos para agrupamiento emplean técnicas de optimización para obtener grupos que maximizan la similitud entre los elementos dentro de cada grupo y minimizan la similitud entre elementos de grupos distintos, tal y como se define la tarea de agrupamiento. Finalmente, cerraremos esta sección con casos prácticos de aplicación de los algoritmos evolutivos para resolver problemas de agrupamiento.

3.1. Algoritmos evolutivos

Los algoritmos evolutivos surgen a finales de los años 60 y son algoritmos estocásticos, inspirados en el proceso de la evolución humana [Holland, 1975] [Goldberg, 1989]. Se han aplicado de forma satisfactoria en problemas reales para optimizar y obtener soluciones en problemas complejos. Su funcionamien-

to es el siguiente [Eiben et al., 2015]: comienzan con una población inicial de posibles soluciones (individuos) que han sido generadas de forma aleatoria mediante una representación genética. Los individuos de la población, conocidos como cromosomas, son evolucionados con el tiempo mediante una función de aptitud (fitness). La función aptitud determina si un individuo es candidato como solución al problema. Aquellos individuos con mejor aptitud son candidatos para ser seleccionados. Sin embargo, no siempre se selecciona los mejores individuos, sino que en algunos casos es recomendable seleccionar aquellos que no tienen la mejor aptitud para mantener la diversidad. Entonces, los individuos seleccionados se reproducen utilizando operadores genéticos de cruce y mutación para generar la nueva población. Esta nueva población reemplaza a la anterior. Este proceso se repite hasta un número máximo de evaluaciones de individuos o alcanzar una solución óptima de la función de aptitud. Los siguientes cinco elementos son esenciales en la aplicación de los algoritmos evolutivos:

- Una representación genética de las posibles soluciones del problema.
- Un enfoque para crear una población de posibles soluciones.
- Una función que agrupe las posibles soluciones en función de su aptitud.
- Un método para generar una población de soluciones potenciales (en la mayor parte de los casos, mediante proceso aleatorio).
- Una estrategia para producir una población inicial de soluciones potenciales.

Los aspectos más destacados de los algoritmos evolutivos son:

- Representación de las soluciones: existen diferentes formas de representar las soluciones que dependerán del tipo de los valores y de si éstos están ordenados o no. La forma más común de representar las soluciones es mediante un vector simulando una cadena de material genético, tal y como se presenta en la figura 3.1:

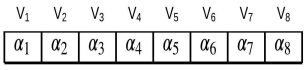


Figura 3.1: Vector que simula una cadena genética. *Fuente: elaboración propia.*

Dentro de las formas de representar las soluciones destacamos:

- Codificación basada en el orden: se aplica para las situaciones donde lo verdaderamente importante es el orden en el que aparecen las variables que identifican la solución.

- Codificación no basada en el orden: se utiliza en los casos en los que el orden de los elementos en las soluciones no es importante, ya que lo que realmente importa es el valor de cada elemento.

    - Codificación binaria: representa las soluciones como una cadena de bits, cuyos valores posibles son 0 o 1. Cada posición de la cadena de bits puede corresponder a una característica o parámetro de la solución. Los algoritmos evolutivos utilizan esta codificación para realizar operaciones como cruce y mutación en la búsqueda de soluciones óptimas, como se verá más adelante. En la figura 3.2 se muestra la representación de una solución mediante codificación binaria.

Figura 3.2: Representación de una solución con codificación binaria. *Fuente: elaboración propia.*

    - Codificación entera: el contenido de las soluciones se representa con valores que pertenecen al conjunto de los números $\mathbb{Z}$. La representación de una solución codificada mediante valores enteros se muestra en la figura 3.3.

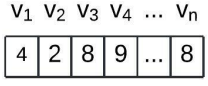


Figura 3.3: Representación de una solución mediante codificación entera. *Fuente: elaboración propia.*

- Codificación real: consiste en representar las soluciones como secuencias de números de punto flotante. Este tipo de codificación queda reflejado en la figura 3.4, y se usa en problemas de optimización numérica donde las soluciones tienen un rango continuo de valores.

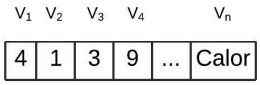


Figura 3.4: Representación de una solución mediante codificación real. *Fuente: elaboración propia.*

- Mecanismo de selección: se encarga de seleccionar qué individuos van a poder reproducirse y cuáles no. Para ello, se debe seleccionar aquellos individuos que son más aptos para reproducirse. Los individuos menos aptos deben mantenerse con el objetivo de no alcanzar poblaciones homogéneas en pocas generaciones. Existen diferentes mecanismos para la selección como son: selección por ruleta, torneo, rango lineal y elitismo entre otros [Eiben et al., 2015]. La elección del mecanismo depende en gran medida del problema a resolver. La forma en que los algoritmos evolutivos seleccionan los individuos es copiando los cromosomas de una generación a la siguiente, donde el número de copias de cada cromosoma depende de la función de aptitud.

- Operador de cruce: en esta fase se realiza un proceso de combinación donde se mezclan los cromosomas de los individuos con la finalidad de obtener individuos diferentes a los ya existentes. La operación de cruce solo se realiza sobre una parte de la población, en función de la probabilidad de cruce. En el caso de que todos los individuos se crucen surge un problema de convergencia prematura, donde la población pierde diversidad y limita la capacidad de encontrar soluciones óptimas. Existen distintos tipos de operadores de cruce: cruce en un punto, en dos puntos, uniforme, aritmético y de ciclo, entre otros. La figura 3.5 es un ejemplo de cruce en un punto, donde aleatoriamente se marca una posición de corte en los cromosomas padres para combinarse.

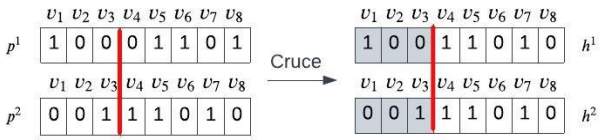


Figura 3.5: Operación de cruce en un punto. *Fuente: elaboración propia.*

- Operador de mutación: consiste en la incorporación de cambios aleatorios en las características de los cromosomas de un individuo. Estos cambios persiguen obtener diversidad en la población, evitando la obtención de resultados prematuros con soluciones poco óptimas. La tasa de mutación indica la probabilidad con la que se mutará una población. La figura 3.6 muestra un ejemplo de mutación mediante la alteración aleatoria del valor de v_3.

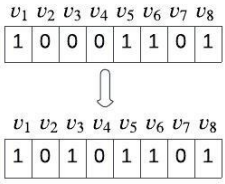


Figura 3.6: Representación de un operador mutación. *Fuente: elaboración propia.*

- Modelos de población: es otro de los aspectos a destacar en el proceso evolutivo. En las diferentes referencias existentes en la literatura destacan dos modelos:

  - Generacional: consiste en crear una población con nuevos individuos por cada iteración. En este caso la nueva población reemplaza directamente a la antigua.
  - Estacionario: consiste en escoger dos padres de la población de cada iteración y aplicarle los operadores genéticos. Los descendientes obtenidos reemplazan a los individuos de la población inicial.

3.2. Algoritmos evolutivos de agrupamiento

Los algoritmos de agrupamiento tienen como objetivo descubrir patrones y establecer relaciones entre los datos. Durante décadas, se ha considerado que el agrupamiento es una herramienta clave para abordar problemas complejos. En la búsqueda del conjunto óptimo de grupos, se requiere calcular las distancias mínimas de todos los datos a los centroides seleccionados inicialmente. Sin embargo, este proceso es computacionalmente costoso y a menudo no es factible resolverlo en un tiempo razonable. Además, la elección adecuada del algoritmo de agrupamiento es un desafío importante, ya que encontrar el algoritmo óptimo para un conjunto de datos específico puede ser complicado. Otro desafío significativo es determinar el número óptimo de grupos. En el caso de los algoritmos de agrupamiento particional, esto puede resultar especialmente difícil, ya que son altamente sensibles a la elección de centroides, lo que puede llevar a resultados poco óptimos si no se seleccionan adecuadamente. En la literatura los problemas de agrupamiento son formulados como problemas de optimización de objetivo único [Priya et al., 2020], destacando tanto algoritmos deterministas como estocásticos como los más populares.

- Los algoritmos deterministas siguen siempre la misma estrategia a la hora de resolver el problema, por lo tanto, siempre devuelven la misma solución al problema con los mismos datos de entrada.
- Los algoritmos estocásticos incluyen elementos aleatorios o incertidumbre que pueden provocar diferentes resultados para los mismos datos.

Los algoritmos estocásticos tienen la capacidad de generar soluciones óptimas globales más eficientes, aunque no siempre están garantizadas. Un ejemplo de algoritmos estocásticos son los metaheurísticos. Los algoritmos metaheurísticos son algoritmos de búsqueda de alto nivel ya sea local y global, que permiten obtener soluciones de alta calidad en un tiempo razonable [X.-S. Yang, 2010]. Los algoritmos inspirados en la naturaleza entran dentro del ámbito de algoritmos metaheurísticos, que son considerados pioneros en la optimización de problemas complejos. Los algoritmos evolutivos son un tipo de metaheurísticas inspirada en la naturaleza. Tratan de resolver los principales problemas de estancamiento en óptimos globales o la convergencia hacia óptimos globales. Los algoritmos evolutivos se comprometen a la diversificación (búsqueda a escala global) y la intensificación (búsqueda a escala local), lo que evita estancarse en los óptimos locales. Tradicionalmente, los algoritmos evolutivos dependen de una población

inicial de soluciones que van incorporando nuevos elementos de forma iterativa que sustituyen a los peores, para compartir información durante el proceso de búsqueda. Los algoritmos evolutivos más destacados son:

- Algoritmo genético (*Genetic algorithm*, GA) [Holland, 1975]: es uno de los algoritmos de optimización más antiguo. Imita el proceso natural basado en la teoría de Darwin sobre la evolución de las especies. Los GA parten de una población de individuos, donde cada uno de ellos representa una solución. Los individuos reciben el nombre de cromosomas. El problema a resolver viene definido por una función objetivo, por lo que el ajuste del individuo a una función objeto se realiza mediante la asignación de un valor que representa la calidad. Este valor se conoce con el nombre de aptitud y es unos de los principales factores en la evaluación. Los GA utilizan esencialmente tres tipos de operadores: selección, cruce y mutación. La selección genera una nueva población basada en los valores de aptitud de los individuos generados en la población anterior. El operador de cruce intercambia partes de dos individuos seleccionados. Y finalmente el operador de mutación altera determinados genes de un individuo.

- Evolución diferencial (*Differential Evolution*, DE) [H. Li et al., 2016]: es un tipo específico de algoritmo evolutivo utilizado en tareas de optimización global de funciones no lineales y no derivables. La ventaja de este tipo de algoritmo respecto a los GA, es que son más fáciles de usar, hacen un uso más eficiente de la memoria, y suponen un menor coste y esfuerzo computacional. Además, una de las principales diferencias del algoritmo DE es que está centrado en el cálculo de la diferencia entre dos individuos de una población elegidos al azar. El algoritmo DE evita atascos en un extremo local de la función de optimización [Slowik, 2010].

- Programación genética (*Genetic programming*, GP): es una técnica de programación evolutiva inspirada en la evolución natural y genética. Es más moderno que los GA y utilizan operadores genéticos modificados. Fue implementado por [Koza, 1994] con el objetivo de encontrar el camino hacia la generación automática de código a partir del conocimiento de criterios de evaluación. Dado que el objetivo era encontrar un programa, los posibles objetivos se codifican en forma de árboles en lugar de cromosomas (bits o números) como tradicionalmente funciona GA. GP difiere de los

GA en el esquema de codificación utilizado. Los operadores genéticos están especializados para trabajar con árboles, el cruce como intercambio de subárboles y la mutación en el intercambio de nodo u hoja.

En base a la definición de los algoritmos mencionados anteriormente, se pueden establecer dos grandes tipos de algoritmos: por un lado, los GA y DE que pertenecen al grupo de los evolutivos, y el resto, que pertenecen al grupo de inteligencia de enjambre. Estas dos clasificaciones comparten características comunes como son: el punto inicial donde se inicializa aleatoriamente el tamaño de la población, y en segundo lugar la identificación de los candidatos para representar la solución [Ezugwu et al., 2021].

Los GA se utilizan en problemas de agrupamiento para encontrar un número óptimo de grupos, con el objetivo de que el agrupamiento sea el más apropiado. Para ello los operadores de selección de las técnicas de agrupamiento basadas en GA tratan de controlar la dirección de la búsqueda, mientras que los operadores de cruce y mutación generan nuevas regiones para la búsqueda. En el algoritmo 1 se muestran los pasos básicos de los GA, que también se siguen en los GA para agrupamiento.

Algoritmo 1 Pasos básicos en un GA.

1: $t = 0$
2: Inicializar la población $P(t)$
3: Calcular la función aptitud $P(t)$
4: $t = t + 1$
5: Si se cumple el criterio de terminación, ir al paso 10
6: Seleccionar $P(t)$ de $P(t-1)$
7: Cruce $P(t)$
8: Mutación $P(t)$
9: Volver al paso 3
10: Seleccionar la mejor salida y parar

A continuación se detallan estos pasos:

- Inicializar la población: los centros de los k grupos codificados en cada cromosoma se inicializan seleccionando aleatoriamente k puntos del conjunto de datos. Este procedimiento se repite para cada uno de los P cromosomas de la población, donde P representa el tamaño de la población.

- Calcular la función aptitud [Maulik et al., 2000]: se lleva a cabo de la siguiente manera. Los grupos se forman en función de los centros codificados en el cromosoma indicado. Esto se hace asignando cada punto x_i, $i = 1, 2, \ldots, n$, a uno de los grupos C_j con centro z_j, tal como se indica (Ecuación 3.1):

$$\|x_i - z_j\| < \|x_i - z_p\|, \quad p = 1, 2, \ldots, k, \quad y \quad p \neq j \tag{3.1}$$

  En los casos donde se produzca un empate, se resuelve de forma aleatoria. Después de la agrupación, los centros de los grupos codificados en el cromosoma son reemplazados por los puntos medios de los grupos correspondientes. Esto se traduce en que para el grupo C_i, el nuevo centro z_i^* es calculado como (Ecuación 3.2):

$$z_i^* = \frac{1}{n_i} \sum_{x_j \in C_i} x_j, \quad i = 1, 2, \ldots, k \tag{3.2}$$

- Selección: en este proceso se seleccionan los cromosomas de la reserva de apareamiento siguiendo el concepto de supervivencia del más apto de los sistemas genéticos naturales. Para ello, a cada cromosoma se le asigna un número de copias proporcional a la función aptitud. Entre otros métodos, se puede utilizar la selección por ruleta.

- Cruce: es un proceso probabilístico que intercambia información entre dos cromosomas parentales para generar dos cromosomas descendientes. Si, por ejemplo, se utiliza el cruce en un punto con una probabilidad de cruce μ_c con cromosomas de longitud l, se genera un número entero aleatorio denominado punto de cruce en el intervalo $[1, l - 1]$. Las partes de los cromosomas situadas a la derecha del punto de cruce se intercambian para producir dos descendientes.

- Mutación: en el proceso de mutación los genes de un cromosoma son modificados con una probabilidad fija μ_m. Por ejemplo, para el caso de representación binaria de los cromosomas, una posición de un bit se muta simplemente invirtiendo su valor.

- Condición de parada: los procesos de cálculo de la función aptitud, selección, cruce y mutación se ejecutan durante un número máximo de iteraciones. La mejor solución obtenida a lo largo de las generaciones mediante funciones elitistas representa la solución al problema. Al final de este proceso, podemos mantener los centroides de los grupos finales.

En la literatura se han propuesto una gran variedad de algoritmos de agrupamiento basados en GA. Un ejemplo es el algoritmo HG-Means [Gribel et al., 2019] que es un híbrido de un GA encargado de reducir la suma del error al cuadrado realizando amplios análisis computacionales para medir la correlación entre la calidad de la solución y el rendimiento del agrupamiento. [José-García et al., 2016] propone estimar el número de grupos de forma automática basado en GA mediante la codificación basada en centroides de longitud variable, longitud fija, en etiquetas y basado en binarios. El algoritmo FPAGA es un híbrido entre el algoritmo de polinización de flores PFA y un GA. Son utilizados de forma conjunta para diversificar el espacio de búsqueda de la solución y mejorar su capacidad.

3.3. Aplicación de algoritmos evolutivos para agrupamiento en diferentes campos

En esta sección se detallan los diferentes sectores donde se han aplicado de forma satisfactoria los algoritmos evolutivos para optimizar problemas de agrupamiento. Además, se incluyen otros algoritmos de optimización como ACO, FA y PSO, ya que comparten características comunes con los algoritmos evolutivos.

Los sectores más destacados son:

- Bioinformática y Salud: en [Capor-Hrosik et al., 2019], los autores propusieron un híbrido de FA y el algoritmo de agrupación k-means para la detección de tumores cerebrales, segmentando la imagen cerebral mediante la función de aptitud del FA y la optimización de la compactación mediante k-means para buscar los centroides óptimos. En [Lai et al., 2009], los autores utilizaron enfoques evolutivos para el agrupamiento jerárquico en la segmentación de imágenes médicas. [Marghny et al., 2011] propuso un enfoque evolutivo para agrupar datos de Hepatitis-C. [Ju et al., 2016] se anticipa a la integración de los EAs multi-objetivo para la agrupación de redes complejas, proponiendo un EA multi-objetivo que trata de agrupar redes de genes con enfermedades y la identificación de redes de proteínas de unión al ADN. Los autores de [Saha et al., 2018] diseñaron dos enfoques de optimización multi-objetivo basados en PSO y DE para agrupar conjuntos de datos de expresión génica en el contexto de la clasificación del cáncer. Se utilizó un algoritmo de agrupamiento evolutivo mejorado llamado iECA* [Hassan et al., 2021] para agrupar eficazmente COVID-19

y conjuntos de datos de enfermedades médicas. El algoritmo se comparó con los algoritmos más avanzados y demostró un mejor rendimiento en la agrupación de conjuntos de datos de enfermedades médicas y un menor tiempo de ejecución y consumo de memoria. Los conjuntos de datos médicos tienen características particularmente complejas que pueden hacer que los algoritmos de agrupación tradicionales tengan un resultado pobre en entornos de *Big Data*, por lo que es posible hacer uso de un nuevo algoritmo de agrupación basado en un método evolutivo inmune modificado bajo la nube.

- Búsqueda en la web y recuperación de información: [Jianrui Chen et al., 2018] desarrolló un método de agrupación heterogénea de EA para predecir la calificación de un enfoque de filtrado colaborativo. [Kushwaha et al., 2018], implementa un PSO binario para la agrupación de textos a gran escala y se utiliza para realizar la selección de características. [Priya et al., 2020] propuso un algoritmo de agrupamiento evolutivo basado en aspectos, un tema con un papel importante en la minería de opiniones. [Said et al., 2018] diseñó un AG basado en agrupamiento para la detección de comunidades en redes sociales, utilizando una métrica de modularidad para cuantificar la calidad de los grupos. [Abualigah et al., 2018] propuso un método para la selección de características utilizando PSO para el agrupamiento de documentos. [Song et al., 2015] discute en un enfoque evolutivo híbrido para el agrupamiento de documentos de texto.

- Inteligencia empresarial y seguridad: [Chou et al., 2017] propuso un híbrido de AG y C-means difuso para la predicción de quiebras, en el que C-means difuso se integra como función de aptitud para buscar el mejor conjunto de características que mejore la precisión de predicción del AG. [Berbague et al., 2018] propone un enfoque de agrupación evolutiva para mejorar el procedimiento de los sistemas de recomendación que combina un AG con k-means y utiliza como función de aptitud la suma de la precisión de grupo y la diversidad de centros. [Barros Franco et al., 2018], introduce un enfoque evolutivo con Fuzzy C-means para la agrupación de la energía solar, donde PSO, DE, y GA fueron comparados para obtener el mejor agrupamiento. [G. Wang et al., 2010] desarrolla un enfoque híbrido de redes neuronales artificiales y agrupamiento difuso para mejorar la eficiencia de los sistemas de: detección de intrusos, reducir su tasa de falsas alarmas y mejorar la seguridad de los servicios computacionales. [Kaur et

al., 2017] presentó un híbrido de FA y k-means para la detección de intrusos en el que el algoritmo FA se utiliza para mejorar la lenta convergencia de k-means.

- Procesamiento de imágenes y reconocimiento de patrones: en este campo, los algoritmos de agrupamiento evolutivo se utilizan principalmente para identificar regiones de especial interés en una imagen. Es posible segmentar una imagen utilizando un modelo de agrupamiento difuso no supervisado basado en un algoritmo evolutivo para obtener información y su posterior etiquetado [Mengxuan Zhang et al., 2019]. Los autores [Bandyopadhyay et al., 2002] aplicaron el agrupamiento evolutivo para distinguir regiones paisajísticas como ríos, viviendas y áreas de vegetación en imágenes de satélite. En una aplicación similar [H. Liu et al., 2015], se emplea la agrupación evolutiva para la extracción automatizada de carreteras a partir de imágenes de satélite. [Omran et al., 2005] aplicó un PSO particularmente en imágenes sintéticas de resonancia magnética y de satélite para buscar el conjunto óptimo de centroides para un número predefinido de grupos. [Y. Han et al., 2007], implementó un enfoque de agrupamiento difuso basado en ACO para la segmentación de imágenes. [Caron et al., 2018] propuso un enfoque de agrupamiento profundo de k-means y redes neuronales convolucionales para aprender y agrupar características visuales, utilizando los conjuntos de datos ImageNet e YFCC100M para las pruebas. En [Cerreto et al., 2018], se aplicó un enfoque de agrupamiento para el reconocimiento de patrones de retrasos ferroviarios. [Feller et al., 2018] aplicó un enfoque de agrupamiento jerárquico para reconocer patrones clínicamente útiles de conjuntos de datos generados por pacientes. En [Hall et al., 1999], los autores aplicaron una estrategia de agrupación difusa guiada genéticamente a la cuantificación de imágenes de resonancia magnética de tejido cerebral.

- Red de transferencia de datos: dentro del campo de las telecomunicaciones, es posible aplicar algoritmos evolutivos para la detección de comunidades en redes dinámicas utilizando algoritmos evolutivos basados en la agrupación espectral para identificar comunidades de nodos basados en la conexión entre ellos [Karaaslanlı et al., 2021]. La red móvil ad-hoc (MANET) es una red autónoma con el problema de que la agrupación de nodos y el encaminamiento pueden llegar a ser complejo y presentar problemas de

seguridad. [Selvakumar et al., 2022] propone un agrupamiento energético eficiente con un protocolo de enrutamiento seguro denominado EECSRP que utiliza algoritmos evolutivos híbridos para MANET.

4

Sistemas difusos evolutivos para agrupamiento múltiple en entornos complejos de flujos continuos de datos

En esta sección, presentamos diversas propuestas basadas en sistemas evolutivos que optimizan el agrupamiento de datos. Las propuestas se basan en crear grupos compactos y eficientes, gestionar la incertidumbre y la imprecisión a la hora de agrupar los datos y ofrecer la mejor solución al problema desde diversas perspectivas. Para ello, esta sección se organiza de la siguiente manera:

- En la sección 4.1, se exponen diferentes conceptos teóricos que son de especial relevancia para entender las propuestas desarrolladas.
- La sección 4.2, presenta una propuesta basada en el algoritmo evolutivo CHC, utilizado en la optimización de problemas complejos. Este algoritmo se combina con los hiper-rectángulos para mejorar la división del conjunto de datos en el espacio. La propuesta resultante recibe el nombre de *CHCClust*.

- Continuamos con la sección 4.3, donde se presenta el algoritmo *MultiCHC-Clust* [Pérez-Martos et al., 2023], un algoritmo de post-procesamiento que es utilizado con el objetivo de agrupar los datos teniendo en cuenta las propuestas de otros agrupamientos para seleccionar la mejor solución de entre todas las posibles.

- Finalmente cerramos con la sección 4.4, donde presentamos una propuesta llamada *FuzzyMultiCHCClust-DS* [Pérez-Martos et al., 2023] que trata de optimizar los agrupamientos resolviendo los problemas de incertidumbre en entornos complejos de flujos continuos de datos.

4.1. Conceptos teóricos

En la subsección 4.1.1, se presenta la lógica difusa, que es utilizada para manejar la ambigüedad propia de la naturaleza de los datos y mejorar la flexibilidad en la asignación de los datos a los diferentes grupos. Por otro lado, en la subsección 4.1.2, se introduce la minería de flujos continuos de datos, un enfoque emergente que permite el análisis en tiempo real y la toma de decisiones de datos que son recibidos de manera constante. Finalmente, en la subsección 4.1.3, se exponen los hiper-rectángulos que tratan de dividir el espacio de datos en regiones con forma de rectángulos, mejorando la distribución de los datos en grupos. Además se definen diferentes métodos para resolver solapamientos entre hiper-rectángulos mediante ajustes o divisiones.

4.1.1. Lógica difusa

La lógica difusa es un campo innovador en la teoría de la computación y tuvo su primera referencia con Lotfi A. Zadeh, quien introdujo una técnica basada en valores lógicos multivaluados [Zadeh, 1965]. Zadeh marcó el inicio de la teoría de los conjuntos difusos, sentó las bases de lo que luego se conocería como lógica difusa e introdujo el concepto de conjuntos difusos, que permiten representar la imprecisión y la vaguedad presente en muchas situaciones del mundo real. En contraste con los conjuntos convencionales, donde un elemento pertenece o no pertenece al conjunto de manera nítida. Este enfoque innovador abrió nuevas perspectivas en el campo de la inteligencia artificial, el control de sistemas y la toma de decisiones, ofreciendo una manera más flexible y realista de modelar y manejar la incertidumbre.

Los conjuntos difusos nos permiten representar y analizar eficazmente conceptos o variables complejas que no tienen límites claros. Gracias a esta capacidad de expresión de los conjuntos difusos podemos simplificar tanto las reglas como los sistemas que se fundamentan en ellas. La división de conjuntos difusos en subconjuntos difusos más pequeños y específicos se conoce como partición difusa. La partición difusa de una variable consiste en dividir el rango de valores de la variable en diferentes conjuntos difusos, permitiendo solapamientos en las fronteras. Esto es similar a cómo funciona el razonamiento humano cuando manejamos valores lingüísticos [Zadeh, 1975]. Se entiende por variable lingüística aquella cuyos valores son palabras u oraciones en un lenguaje natural o artificial. Por ejemplo, la edad es una variable lingüística si sus valores son lingüísticos y no numéricos (por ejemplo, Bajo, Normal, Alto y Muy Alto). Las variables lingüísticas se refieren a la idea o al concepto que vamos a calificar como difuso (por ejemplo, Temperatura, Edad, Altura). La importancia de los términos viene definida por una función de pertenencia y por un conjunto difuso. La función de pertenencia indica el grado de pertenencia del valor de una variable a un conjunto difuso.

Definición 3 *Los valores de una variable del conjunto difuso* A *pueden definirse mediante la función de pertenencia, la cual se denota mediante:*

$$\mu_A(x) \in [0, 1] \tag{4.1}$$

donde x *es el valor de una variable lingüística.*

Un ejemplo de particionamiento difuso se puede observar en la figura 4.1 donde tenemos una variable con tres etiquetas lingüísticas con valores de las etiquetas Bajo, Normal, Alto. Además tenemos funciones de pertenencia triangular y trapezoidal. La función triangular es ampliamente utilizada para representar conjuntos con un grado de pertenencia gradual. Esta función asigna un valor máximo de pertenencia a un punto central y disminuye de manera lineal hacia ambos extremos. Está definida por tres parámetros: el valor mínimo, el valor máximo y el punto medio. Este tipo de función es particularmente útil en contextos donde es más probable que la pertenencia se encuentre en un rango específico de valores cercanos al punto central. La función trapezoidal comparte similitudes con la función triangular al asignar un grado de pertenencia máximo a un rango de valores, en lugar de a un único valor central. Esta función se define por cuatro parámetros distintivos: el valor mínimo, el máximo, el punto medio

y la pendiente de las dos ramas laterales. Se utiliza en contextos donde es probable que la pertenencia se encuentre predominantemente dentro de un intervalo específico de valores.

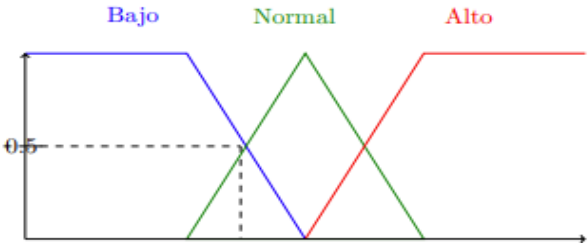


Figura 4.1: Ejemplo de particionamiento difuso con una variable lingüística con tres etiquetas. *Fuente: elaboración propia.*

Los sistemas que se apoyan en reglas y emplean conjuntos difusos para representar los valores de sus variables son conocidos como sistemas basados en reglas difusas. Una regla se puede aplicar si su condición de inicio (antecedente) coincide con la zona del espacio donde está el ejemplo y el grado de pertenencia es mayor que cero entre la condición de inicio y el ejemplo. El grado de pertenencia se obtiene de la siguiente manera:

- Se calcula el grado de pertenencia de cada una de las variables al conjunto difuso correspondiente.
- Usualmente, el inicio de una regla difusa se compone de varias condiciones que deben cumplirse para las diversas variables. En este caso, los grados de pertenencia se combinan mediante el uso del operador T-norma.
- Cuando las condiciones para las variables en el inicio de una regla difusa se combinan mediante el operador de disyunción, también se empleará el operador de unión, utilizando una T-conorma para calcular el grado de compatibilidad.

Hay varias maneras de representar las reglas en los sistemas que las usan. Los sistemas más comunes para representar reglas son la representación canónica y la representación en forma disyuntiva (DNF, por sus siglas en inglés). La representación canónica se basa en la conjunción de pares atributo-valor, mientras que la representación disyuntiva es similar pero permite más de un valor para cada atributo.

Un ejemplo de regla difusa canónica sería:

$$\text{R1:} \quad \text{SI } X_0 = \text{Bajo}_0 \text{ Y } X_4 = \text{Normal}_6 \text{ ENTONCES Bajo-Coste}_5$$

En dicho ejemplo podemos observar la asignación de un único valor a cada variable, donde todas las variables son unidas mediante el operador Y. Para el caso de representación de reglas difusas en forma disyuntiva se realiza de la siguiente manera:

$$\text{R1:} \quad \text{SI } X_0 = (\text{Bajo}_0 \text{ O Normal}_0) \text{ Y } X_4 = (\text{Normal}_6 \text{ O Alto}_6) \text{ ENTONCES Bajo-Coste}_5$$

Por lo tanto, el poder interpretar el conocimiento nos va a permitir extraer un conocimiento de calidad. La forma de hacerlo es mediante sistemas basados en reglas que son los más destacados dentro de la lógica difusa.

4.1.2. Minería de flujos continuos de datos

En la última década se ha producido un importante avance tecnológico con la aparición de dispositivos que generan grandes cantidades de datos, como dispositivos inteligentes, redes sensoriales, dispositivos médicos, vídeos y redes sociales entre otros. Estos datos son generados con una frecuencia elevada, dando lugar a grandes volúmenes de datos que son tratados como flujos continuos de datos. Estos flujos de datos son potencialmente infinitos y su almacenamiento, procesamiento y tratamiento requiere de la aplicación de métodos distintos a los tradicionales. Analizar y tomar decisiones de forma rápida y eficaz a partir de un flujo de información se le conoce con el nombre de minería de flujo de datos [Anjum et al., 2024]. La minería de flujos de datos es considerada como un subcampo de la minería de datos, el aprendizaje automático y el descubrimiento de información. Difiere de la minería de datos en que los datos son recibidos de forma constante y cuya finalidad es detectar posibles patrones, tendencias o anomalías que pueden ser útiles para la toma de decisiones instantáneas. Dado que el almacenamiento de los flujos es imposible debido a su volumen, la mayor parte de los algoritmos se limitan a leer una única vez o un pequeño número de veces dependiendo de la capacidad del sistema. La manera de gestionar este volumen de datos puede ser [Ramzan et al., 2023]:

- Online: en este punto las instancias llegan de manera secuencial y son procesadas de forma individual por el algoritmo.

- En bloque: las instancias se van almacenando de forma secuencial hasta crear un bloque de un tamaño predeterminado que se procesa en conjunto.

4.1.3. Rectángulos de N dimensiones: Hiper-rectángulos

En esta sección exploraremos una técnica para agrupar datos que difiere notablemente de los métodos utilizados por los algoritmos convencionales de agrupamiento. Además de presentar esta técnica y sus características distintivas, analizaremos las múltiples ventajas que ofrece. Concluiremos esta sección con recomendaciones para resolver los posibles conflictos que puedan surgir durante el agrupamiento de los datos.

Concepto

El agrupamiento de los datos depende en gran medida de la estrategia seleccionada, la medida utilizada para el cálculo de las distancias, así como el cálculo de los centroides. Otra de las formas de poder distribuir los datos en grupos es mediante hiper-rectángulos. Un hiper-rectángulo, también conocido como hipercaja, es una figura geométrica en un espacio de D-dimensiones, equivalente a un rectángulo en un plano bidimensional o a un cuboide en un espacio tridimensional.

Aunque el hiper-rectángulo tiene la capacidad de rotar alrededor de cada uno de sus ejes, las caras deben ser paralelas a dichos ejes para su aplicación en el agrupamiento. En situaciones específicas, la definición de un hiper-rectángulo H en un espacio de D-dimensiones implica la especificación de los valores mínimos y máximos a lo largo de cada dimensión, que son esencialmente los ejes en el espacio.

Definición 4 *Sea* H *un hiper-rectángulo, se establece dos conjuntos de datos. El conjunto de valores mínimos* $H_n = \{h_{n_i}, \forall i = 1, \cdots D\}$ *y el conjunto de valores máximos* $H_x = \{h_{x_i}, \forall i = 1, \cdots D\}$*, donde* h_{n_i} *representa el límite inferior en la dimensión* i *y* h_{x_i} *el límite superior en* i*.*

Es evidente que, para cualquier hiper-rectángulo en cualquier espacio, se cumple la condición $h_{n_i} < h_{x_i}$, para $i = 1, \cdots D$. Se puede notar que estas representaciones son aplicables únicamente a hiper-rectángulos cuyas caras son paralelas a los ejes. Cualquier rotación en al menos una dimensión invalidaría

estas representaciones. Además, un hiper-rectángulo alineado con los ejes permite la aplicación directa de reglas de clasificación, una posibilidad que se vuelve inviable con un hiper-rectángulo rotado. El uso de los hiper-rectángulos ofrece las siguientes ventajas:

- Trabaja con vectores, reduciendo el proceso computacional de cálculo para operaciones de división, reducción, ampliación o detención de solapamientos.
- Los límites de los hiper-rectángulos son descriptores del conjunto de datos contenido, por lo que es adecuado para extraer conocimiento de manera rápida.

Un ejemplo de representación de hiper-rectángulos en un espacio de dos dimensiones se puede visualizar en la figura 4.2.

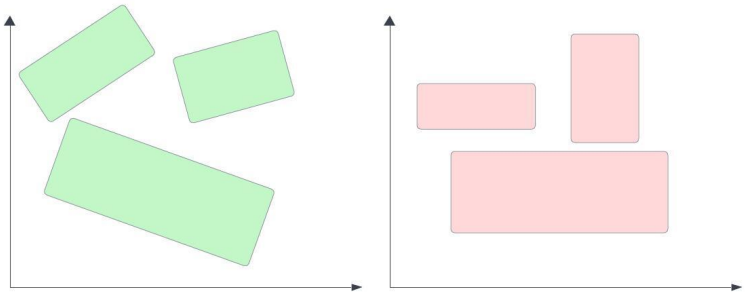


Figura 4.2: Ejemplos de hiper-rectángulos en un espacio de dos dimensiones: hiper-rectángulos rotados respecto a los ejes (izquierda) y rotación paralela respecto a los ejes (derecha). *Fuente: elaboración propia.*

Definición 5 *Dado un objeto* p*, pertenece a un hiper-rectángulo* H *en un espacio D-dimensional si cumple*

$$h_{n_i} \leqslant p \leqslant h_{x_i}, \forall i = 1, \cdots D \tag{4.2}$$

Definición 6 *El número de elementos que están dentro de un hiper-rectángulo* H *se denota como* $\epsilon(H)$.

Dos hiper-rectángulos pueden compartir una región en el espacio, ya sea de manera parcial o completa.

Definición 7 *Dos hiper-rectángulos* H *y* P *muestran intersección en el espacio de dimensión* D *si se satisface al menos una de estas cuatro condiciones en todas y cada una de las dimensiones:*

1. $H_{n_i} \leqslant P_{n_i} \leqslant H_{x_i}$
2. $H_{n_i} \leqslant P_{x_i} \leqslant H_{x_i}$
3. $P_{n_i} \leqslant H_{n_i} \leqslant H_{P_i}$
4. $P_{n_i} \leqslant H_{x_i} \leqslant H_{P_i}$

Especialmente, si se satisfacen las condiciones 1 y 2 en todas las dimensiones espaciales, P se encuentra completamente contenido dentro de H. Asimismo, si simultáneamente se cumplen las condiciones 3 y 4 en todas las dimensiones, H está totalmente contenido dentro de P.

Tipos de solapamiento

En un conjunto de datos en el que los hiper-rectángulos que representan cada clase no se superponen, no es necesario realizar ajustes, ya que cada clase está representada por su propio hiper-rectángulo. Sin embargo, en situaciones reales es frecuente que los hiper-rectángulos se superpongan. Esto se debe a que un elemento puede estar contenido en más de un hiper-rectángulo y, por lo tanto, puede ser clasificado como perteneciente a varias clases. Resolver estas situaciones no es una tarea trivial. Se pueden llevar a cabo varias acciones como reducir uno o más hiper-rectángulos hasta que el solapamiento sea mínimo (si se aceptan datos falsos positivos) o nulo. Otra posibilidad es dividir los hiper-rectángulos en otros más pequeños. Independientemente de las medidas que se tomen, existen numerosas combinaciones de acciones diferentes para conseguir un modelo de datos que represente un agrupamiento factible, lo que se vuelve más complejo a medida que aumenta el número de dimensiones del espacio de trabajo y el número de clases implicadas. A continuación, detallamos una serie de ejemplos sobre la forma más adecuada de resolver el solapamiento entre datos:

1. **Solapamiento sin datos involucrados**: es el caso más simple de todos. Ocurre cuando se superponen los hiper-rectángulos y no hay datos involucrados, como refleja la figura 4.3. La forma de resolver esta superposición es con la división de uno de los dos hiper-rectángulos solapados, tal y como representa la figura 4.4. De acuerdo con el nivel de superposición entre

hiper-rectángulos se toma la decisión de dividir aquel que está significativamente más implicado. En la figura 4.3 no se ve claramente cuál de los dos está más implicado, por lo tanto, no existe una solución en particular, por lo que la división va a depender del problema. Para problemas donde existe cierta complejidad debido al número de clases e intersecciones, obtener una solución de forma visual puede ser complejo, por lo que se suele considerar que la mejor solución es aquella con menor número de hiper-rectángulos.

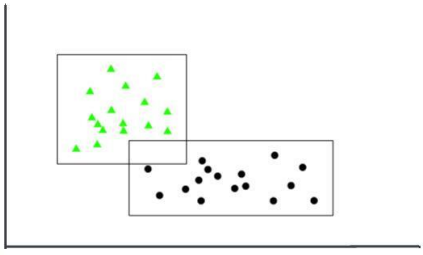


Figura 4.3: Solapamiento sin datos involucrados. *Fuente: elaboración propia.*

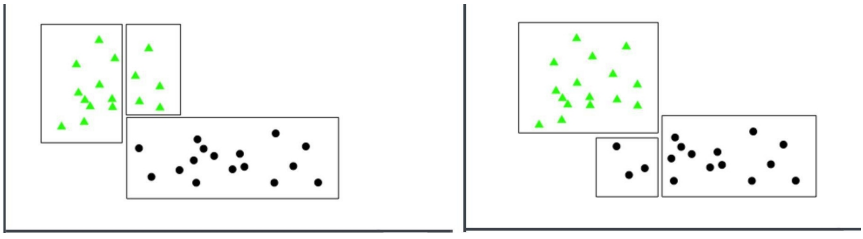


Figura 4.4: Eliminación de la superposición cuando no hay datos involucrados entre dos hiper-rectángulos. *Fuente: elaboración propia.*

2. **Solapamiento con datos de una clase**: en este caso, al proceso es muy parecido al anterior, con la diferencia de que cuando hay datos de una clase implicados en un solapamiento, se debe realizar la división del hiper-rectángulo que no tenga datos de la clase implicada, con el objetivo de que el número de divisiones sea mínimo. La figura 4.5 muestra el solapamiento con datos de una clase.

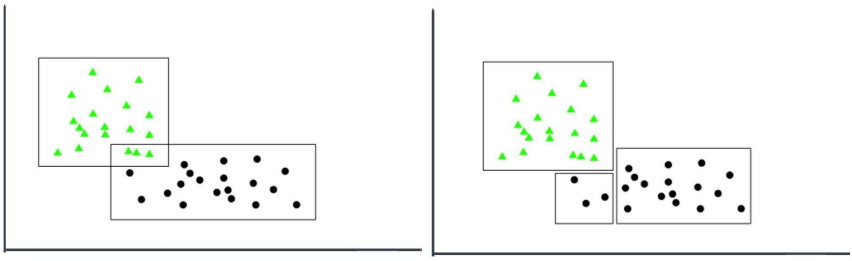


Figura 4.5: Eliminación de la superposición cuando no hay datos involucrados entre dos hiper-rectángulos. *Fuente: elaboración propia.*

En las situaciones donde un hiper-rectángulo H' está incluido en otro hiper-rectángulo H, se observa que cualquier división de H no elimina la superposición, debido a que los nuevos hiper-rectángulos generados por la división seguirán estando dentro de H tal y como aparece en la figura 4.6. Para estos casos se elimina el hiper-rectángulo H y se generan nuevos hiper-rectángulos.

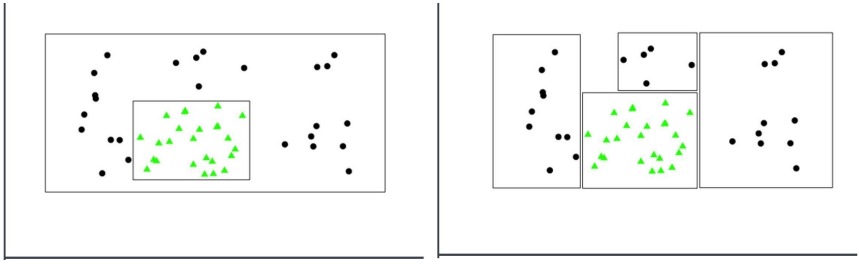


Figura 4.6: Eliminación de la superposición cuando hay datos involucrados de una clase en un hiper-rectángulo. *Fuente: elaboración propia.*

3. **Solapamiento con datos de varias clases**: este tipo de superposición es la más compleja de abordar, ya que la mera división de un hiper-rectángulo hace que los datos de una categoría queden más allá de los límites de sus respectivos hiper-rectángulos, tal y como aparece en la figura 4.7. El problema con este tipo de superposiciones es que no tienen una solución exac-

ta, lo que puede causar variaciones en los resultados. Aunque es posible realizar n divisiones de hiper-rectángulos no se recomienda realizar tantas divisiones ya que obtendría un número elevado de reglas.

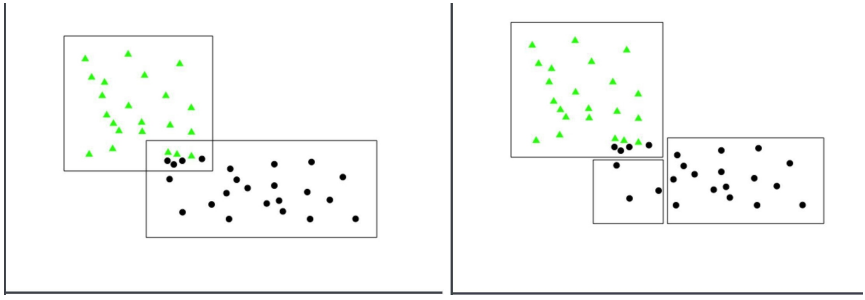


Figura 4.7: Eliminación de la superposición cuando hay datos involucrados de dos clases en un hiper-rectángulo. *Fuente: elaboración propia.*

4.2. Algoritmo *CHCClust*

En esta sección se presenta una propuesta para mejorar la estrategia de agrupamiento y buscar la solución óptima en problemas de agrupamiento. La propuesta está basada en el concepto de hiper-rectángulo y el algoritmo evolutivo CHC. Esta sección está divida en dos partes: en la primera parte se introduce el algoritmo CHC y los elementos innovadores que lo componen, mientras que en la segunda parte se expone como opera el algoritmo propuesto.

4.2.1. Algoritmo CHC

CHC es una innovadora variación del algoritmo genético propuesto por [Eshelman, 1991]. Utiliza un operador genético de recombinación radical, lo que lo convierte en un algoritmo sumamente revolucionario. Además, el algoritmo CHC incorpora una estrategia de selección elitista, asegurando la inclusión exclusiva de los individuos más destacados en la nueva población. Esta combinación de elementos posiciona a CHC como un optimizador que persigue el equilibrio entre variedad y convergencia.

CHC opera bajo la guía de una selección orientada a la supervivencia, también conocida como *selección elitista*, introduciendo un mecanismo que evita la formación de parejas entre individuos que exhiben similitudes. Un componente crucial del algoritmo es el operador genético de recombinación *HUX*, una variante del cruce uniforme con un impacto altamente pionero. Es importante destacar que la etapa de *mutación* no se realiza durante la *recombinación*. La diversidad se mantiene introduciendo variaciones parciales de forma aleatoria cuando la población converge. Este fenómeno, conocido como *reinicialización*, es un aspecto esencial del esquema de CHC, cuya descripción detallada se encuentra en el algoritmo 2.

Algoritmo 2 Pseudocódigo del algoritmo CHC.

Entrada
n: número de individuos reinicialización
$nEval$: número de evaluaciones
r_d: ratio divergencia

$t \leftarrow 0$
$d \leftarrow l/4$ ▷ l: longitud del cromosoma, d: umbral diferencial
$InicPoblacion(P(t))$
$EvaluarPoblacion(P(t))$
while $(t \leqslant nEval)$ **do**
  $t \leftarrow t + 1$
  $C(t) \leftarrow SeleccionarPadre(P(t-1))$
  $C'(t) \leftarrow HUX(C(t))$
  $EvaluarPoblacion(C'(t))$
  $P(t) \leftarrow SeleccionElitista(C'(t), P(t-1))$
  if (no cambios$(P(t), P(t-1))$ **then**
    $d \leftarrow d - 1$
  end if
  if $(d < 0)$ **then**
    $Reinicializar(P(t), n)$
    $Inicializar(d, r_d, l)$
  end if
end while

A continuación, se describen estos cuatro elementos innovadores que incorpora el algoritmo CHC:

- **Selección elitista:** identifica y selecciona los cromosomas más destacados entre padres e hijos, como se indica en la línea 10 del algoritmo 2.
- **Cruce uniforme HUX:** es un operador genético que intercambia posiciones específicas de los cromosomas que son desiguales entre los padres, manteniendo el resto. La cantidad de posiciones intercambiadas es exactamente la mitad. Este operador queda reflejado en la línea 8 del algoritmo 2.
- **Prevención de incesto:** el mecanismo de prevención consiste en calcular la distancia Hamming entre los padres potenciales y, si la mitad de esta distancia no supera el umbral diferencial (d), no se reproducen y no se crea descendencia.
- **Reinicialización:** si d es inferior a cero, entonces la población es reinicializada. Esto se puede llevar a cabo de las siguientes maneras: conversando los mejores o parte de la población y generando el resto de forma aleatorio o seleccionando el mejor elemento como plantilla (con una variación del 35 % [Eshelman, 1991]) e incluyendo una copia del mismo, tal como se describe en la línea 15 del algoritmo 2.

El algoritmo CHC inicia la reproducción después de la inicialización y evaluación de la población. El operador genético *HUX* se encarga de generar descendientes que difieren significativamente de sus padres. En situaciones donde no hay cruce, d se reduce. Si d cae por debajo de cero, se interpreta que la población ha convergido y, en consecuencia, se realiza una reinicialización.

Los parámetros clave que guían la evolución del algoritmo CHC son:

- Umbral diferencial (d): representa el grado máximo de similitud entre cromosomas durante el cruce. Los valores oscilan entre $l/2$ y 0. Destacar que l representa el tamaño del cromosoma. Una vez reinicializado la población se actualiza el valor de d, siendo $l/4$ el valor recomendado.
- Ratio de divergencia (r_d): representa la proporción de bits que deben modificarse en el mejor individuo de la población para generar los restantes individuos durante el proceso de reinicio. Puede tomar valores reales en el intervalo [0,1], siendo 0.35 un valor apropiado.

- Mejores individuos (n): representa los n mejores individuos que permanecerán en la población después del proceso de reinicio. Los restantes serán generados utilizando la ratio de divergencia r_d. Este valor puede ser un número entero entre 1 y el total de individuos que componen la población. No existe un valor recomendado, pero se sugiere que no englobe un gran número de individuos.

4.2.2. Funcionamiento del algoritmo *CHCClust*

CHCClust es una propuesta desarrollada para resolver problemas de agrupamiento. *CHCClust* se construye a partir de la fusión de hiper-rectángulos como solución para mejorar el agrupamiento inicial del conjunto de datos y del algoritmo CHC como optimizador de problemas complejos. Este algoritmo puede ser aplicado sobre los resultados de cualquier algoritmo de agrupamiento. Para entender cómo funciona, se expone el pseudocódigo del algoritmo 3, incluyendo sus parámetros de entrada y salida.

Algoritmo 3 Pseudocódigo del algoritmo *CHCClust*.

Entrada
conjunto: conjunto de datos con los resultados de cualquier algoritmo de agrupamiento
n: número máximo de iteraciones
Salida
resultado: vector con la optimización del conjunto de datos de entrada

```
1: hiper ← CrearConjuntoInicialHiperrectangulos(conjunto)
2: conjunto ← ComponerHiperrectangulos(hiper)
3: t ← 0
4: nEval ← n
5: P_a ← conjunto
6: d ← longitud(conjunto)/4
7: InicializarPoblacion(P_a, d)
8: while t ⩽ nEval do
9:     padres ← SeleccionarPadre(P_a)
10:    hijos ← HUX(P_a(t))
11:    Evaluar(hijos, padres)
12:    P_n ← SeleccionElitista(hijos, padres)
13:    if (no cambios(P_a, P_n)) then
14:        d ← d − 1
15:        if (d < 0) then
16:            P_n ← Reinicializar(P_a)
17:        end if
18:    end if
19:    t ← t + 1
20:    P_a ← P_n
21: end while
22: conjuntoNuevo ← ReducirHiperrectangulos(conjunto, P_a)
23: resultado ← EvarHiperrectangulos(conjuntoNuevo, conjunto)
```

A continuación se explican las instrucciones que componen el algoritmo:

- *Línea 1*: genera inicialmente una serie de hiper-rectángulos a partir del conjunto de datos. Cada dato del conjunto de datos es ubicado en un hiper-rectángulo.
- *Línea 2*: mediante un proceso iterativo los hiper-rectángulos se van fusionando o dividiendo, dependiendo del tipo de solapamiento, hasta llegar a un punto donde ya no haya más cambios. El siguiente paso es aplicar el algoritmo CHC, con la finalidad de reducir y optimizar el número de hiper-rectángulos.
- *Línea 5*: se crea una población inicial de cromosomas basada en la configuración de los hiper-rectángulos. Cada cromosoma cuenta con una cantidad de bits igual a $k * n$, donde k es el número de hiper-rectángulos y n es el tamaño del conjunto de datos. El contenido del cromosoma se completa de manera aleatoria con valores de 1 o 0. La figura 4.8 refleja un ejemplo de inicialización de la población.

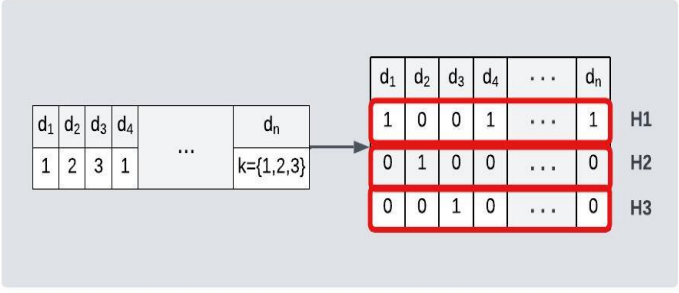


Figura 4.8: Ejemplo de inicialización de una población con $k = 3$ hiper-rectángulos. *Fuente: elaboración propia.*

- *Línea 7*: procede a la inicialización de la población, llevando a cabo el cálculo de la distancia entre cada elemento del conjunto de datos y el hiper-rectángulo, considerando únicamente aquellos casos en los que el bit correspondiente en el cuerpo del cromosoma es 1. El propósito es identificar las instancias en las que la distancia es mínima y la clase asociada al elemento es la correcta. La forma de calcular la distancia (Ecuación 4.3) se lleva a cabo de la siguiente manera:

$$\sum_{i=1}^{n} d[i] = \sum_{j=1}^{n} \left[d[i][j] == 1 \right] \cdot distance(d[i], h[j]) \tag{4.3}$$

donde i representa un elemento del conjunto de datos, d es un vector con el conjunto de datos de entrada, j representa un hiper-rectángulo, h es un vector con los hiper-rectángulos, y el método $distance$ calcula la distancia entre el objeto i y el hiper-rectángulo j.

- *Línea 10 - 11*: el operador de cruce HUX realiza un intercambio preciso de la mitad de las partes del individuo que difieren entre los progenitores. Este proceso asegura que ambos descendientes siempre se ubiquen a la máxima distancia Hamming de sus padres, lo que introduce un nivel significativo de diversidad en la población emergente.
- *Línea 12*: los progenitores de la población actual se unen con la generación de descendientes generada a partir de ellos, eligiendo los mejores individuos para generar la nueva población.
- *Línea 13*: si no se logra generar descendientes que superen la calidad genética de la generación anterior, se procederá a restar 1 al umbral diferencial. Este proceso es comúnmente referido como medida de prevención del incesto.
- *Línea 15*: cuando $d < 0$, la población es reiniciada y se toma al individuo más destacado como el punto de partida del primer cromosoma en la nueva generación. Los cromosomas restantes se generan alterando aleatoriamente un porcentaje (normalmente 35 %) de sus bits.
- *Línea 19 - 22*: se aumenta el contador de iteraciones y asignamos la nueva población. Este procedimiento se repite $nEval$ veces. La etapa final implica la reducción del número de hiper-rectángulos en la población generada por el algoritmo CHC.
- Finalmente, se logra obtener una agrupación del conjunto de datos gracias a la optimización llevada a cabo por el algoritmo CHC sobre los hiper-rectángulos.

Finalizado el proceso de definición de las instrucciones que componen el código, hay que matizar que tenemos un proceso de post-procesamiento que nos permite mejorar la distribución de los elementos en grupos y reducir el número

de grupos para cualquier algoritmo de agrupamiento. El capítulo 5 detalla un estudio experimental utilizando el algoritmo *CHCClust*, el cual pone de manifiesto su efectividad en la tarea de agrupamiento de datos. Los resultados obtenidos revelan la capacidad del algoritmo para generar agrupamientos de calidad para diversos conjuntos de datos.

4.3. MultiCHCClust: un algoritmo evolutivo multi-agrupamiento basado en hiper-rectángulos.

En esta sección, se expone una propuesta innovadora llamada *MultiCHCClust*, que permite agrupar los datos desde diferentes perspectivas. La sección se divide en dos partes: en la primera exploramos el concepto de multi-agrupamiento y sus objetivos; en la segunda se detalla el funcionamiento de *MultiCHCClust*.

4.3.1. Multi-agrupamiento

El algoritmo *CHCClust* destaca por ser un optimizador de los resultados de cualquier algoritmo de agrupamiento. Esta capacidad permite abordar el problema de la estructura del agrupamiento, ya que existen múltiples formas posibles de agrupar los datos, las cuales dependen del algoritmo y de los parámetros de entrada. La existencia de múltiples estructuras posibles fomenta el desarrollo del enfoque conocido como multi-agrupamiento.

El enfoque de multi-agrupamiento representa una evolución del agrupamiento convencional al aspirar a identificar múltiples conjuntos de grupos en lugar de limitarse a sólo uno, como es común en los métodos de agrupamiento tradicionales. Cada partición constituye una agrupación distinta, lo que permite interpretar los datos desde diferentes perspectivas o hipótesis [Yu et al., 2024]. Además, con la cada vez mayor presencia de *Big Data*, la estructura de los datos se está volviendo muy compleja.

Las ideas principales del multi-agrupamiento son:

- Un mismo dato puede pertenecer a diferentes grupos.
- Mejorar la calidad del conocimiento al obtener diversas formas de agrupamiento de los datos.

La figura 4.9 ilustra un caso de multi-agrupamiento, donde un conjunto de datos es particionado en dos agrupaciones distintas. Dentro de cada agrupación, se forman tres grupos, y cada uno de los cuales contiene datos del conjunto original agrupados de forma diferente. Las flechas azules indican cómo un dato puede ser ubicado en diferentes grupos.

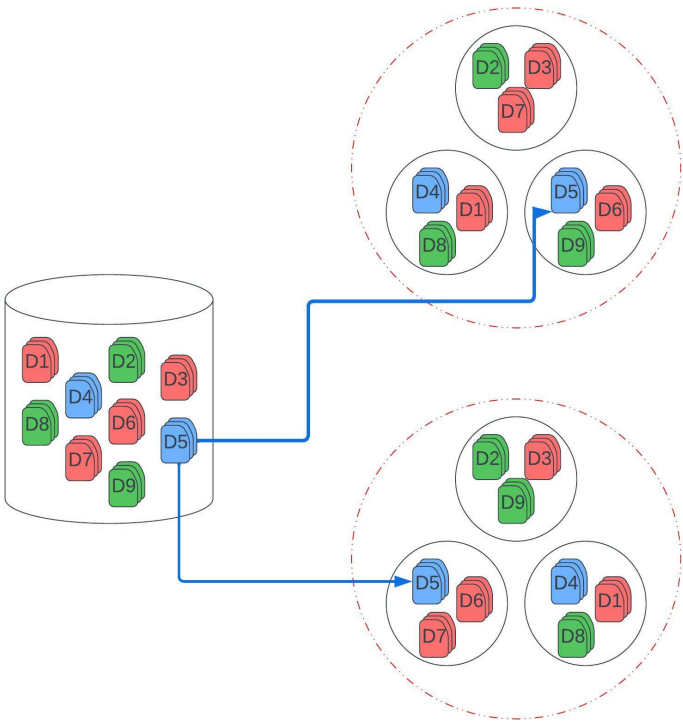


Figura 4.9: Conjunto de datos organizado en diferentes agrupaciones. *Fuente: elaboración propia.*

Cada agrupamiento de la figura 4.9 debería servir como punto de partida para poder generar agrupamientos de calidad.

El multi-agrupamiento puede encontrarse en la literatura bajo diferentes términos, como meta-clustering o ensemble clustering [Caruana et al., 2006] [Boongoen et al., 2018]. La idea principal de estas estrategias es combinar los grupos

para aprovechar el consenso entre varias soluciones de agrupamiento al combinarlas en una única partición que resume de manera óptima el conjunto de datos. Como resultado, se obtiene una solución de agrupamiento que mejora tanto la precisión como la estabilidad con respecto a las soluciones individuales de los algoritmos de agrupamiento [Ayad et al., 2010].

Las funciones de consenso son complementadas con diversas técnicas para evaluar los resultados. Por ejemplo, el esquema de multi-agrupamiento está basado en la selección de modelos [T. Li et al., 2022] que emplean un consenso centrado en una medida de similitud generalizada, que opera desde el nivel de instancia hasta el nivel de grupos. En [Khalili et al., 2021], se sugiere la combinación de un conjunto de algoritmos de consenso con el coeficiente de Jaccard para obtener subconjuntos de particiones jerárquicas con alta diversidad y calidad. Por otro lado, en [B. Zhou et al., 2024], se propone una agregación de medidas de similitud para agrupamiento, donde los datos iniciales son agrupados por consenso utilizando una serie de medidas, con el objetivo de maximizar la similitud entre los grupos resultantes.

El concepto de multi-agrupamiento se ha aplicado en diversas áreas del mundo real. Por ejemplo, en el campo de la electrónica [Mirzaie et al., 2017], se aborda el desafío de seleccionar el nodo óptimo en una red compuesta por múltiples nodos. Sin embargo, este proceso implica una cantidad considerable de mensajes enviados y recibidos debido a la necesidad de realizar varias iteraciones para encontrar el mejor nodo. Para mitigar este problema, se propone un algoritmo de multi-agrupamiento adaptativo basado en lógica difusa [Lingras et al., 2016]. Además, en el ámbito de la minería de datos, donde los datos pueden contener información granular y estar interconectados, la captura de esta información puede resultar desafiante. Para abordar esta dificultad, se sugiere un meta-clustering recursivo. Otra de las áreas donde ha sido aplicado de forma satisfactoria ha sido en la agrupación de genes y la segmentación de imágenes [Ghaemi et al., 2009]. La propuesta se centra en el uso de la distancia geodésica y se materializa en un algoritmo multi-agrupamiento de selección de características basado en mapeo isométrico (MCFS-I) [Y. Wang et al., 2021]. Este enfoque permite la selección adaptativa de características para múltiples grupos sin la necesidad de supervisión.

4.3.2. Funcionamiento del algoritmo *MultiCHCClust*

A partir del concepto de anterior de multi-agrupamiento, proponemos un algoritmo post-procesamiento llamado *MultiCHCClust* que sea capaz de mejorar los resultados obtenidos por el algoritmo *CHCClust*. *MultiCHCClust* recibe como parámetro de entrada los resultados procedentes de cualquier algoritmo de agrupamiento. A estos parámetros, se les aplica el algoritmo *CHCClust* con el propósito de generar soluciones óptimas, en un tiempo razonable, frente a problemas de un alto grado de complejidad. En la fase final del proceso, se lleva a cabo la selección y filtrado de los mejores resultados mediante una función de cribado y consenso.

- Función cribado (Screening en inglés): este tipo de funciones son usadas de forma frecuente para seleccionar o filtrar un conjunto de datos en base a ciertos criterios. El objetivo de esta función es reducir la dimensionalidad del problema, manteniendo al mismo tiempo la información que es más importante o discriminativa.
- Función consenso: se presenta como un enfoque donde se toma una decisión final combinando los resultados o predicciones de múltiples modelos o métodos. Para ello, se recompila y ponderan las predicciones de varios modelos con el objetivo de que la solución no sea única, sino una solución robusta y confiable. La decisión por consenso implica varios pasos:
  - Recompilar las predicciones: en primer lugar, cada uno de los parámetros de entrada produce su propia predicción.
  - Combinación de las predicciones: se combinan las predicciones de los diferentes modelos.
  - Asignación de pesos: de todas las predicciones generadas, el sistema trata de quedarse con las n mejores predicciones.
  - Evaluación y refinamiento: después de la toma de decisión, es importante valorar su rendimiento y si fuese necesario, ajustar o refinar el proceso con el propósito de mejorar el resultado final.

Un ejemplo gráfico sobre el funcionamiento del algoritmo *MultiCHCClust* queda plasmado en la figura 4.10. El algoritmo recibe como entrada un conjunto de resultados que corresponden al agrupamiento llevado por cualquier algoritmo de agrupamiento. Seguidamente, aplica *CHCClust* a cada resultado recibido para optimizar el agrupamiento previo y luego almacenar todos los resultados

en una matriz de resultados. A partir de la información persistida en la matriz, *MultiCHCClust* utiliza una función de cribado para seleccionar y filtrar los resultados midiendo la calidad de la distribución del conjunto de datos en los grupos. La medida empleada para mediar la calidad de los grupos es el coeficiente de silueta. Por último, la función de selección filtrará los n resultados con la mejor distribución para alcanzar posteriormente un consenso sobre los n resultados utilizando un recuento puro. Si no hay consenso en este recuento, *MultiCHCClust* selecciona la mejor opción. El algoritmo devuelve un vector de salida con la nueva agrupación obtenida de la función de consenso. *MultiCHCClust* es capaz de indicar el consenso completo, consenso parcial ($*$) y no consenso ($**$).

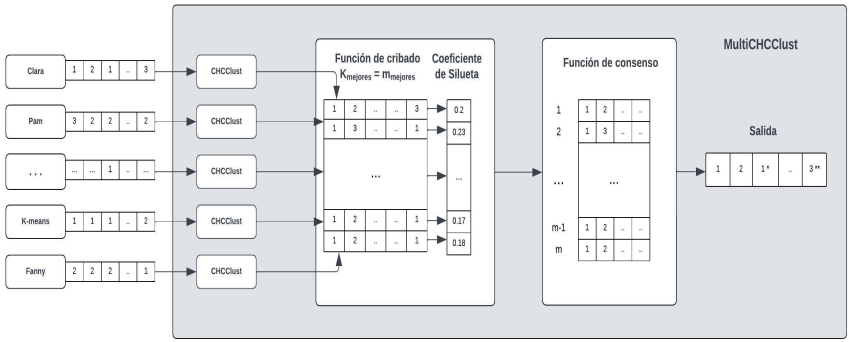


Figura 4.10: Esquema operacional del algoritmo *MultiCHCClust. Fuente: elaboración propia.*

De esta forma, el algoritmo *MultiCHCClust* aporta las siguientes ventajas:

- Mejora de la robustez: al combinar los resultados de múltiples algoritmos de agrupamiento, tiende a producir particiones más estables. Esto significa que la agrupación resultante es menos sensible a pequeñas variaciones en los datos de entrada o en los parámetros del algoritmo.

- Mayor precisión: al aprovechar la diversidad de múltiples algoritmos, puede capturar diferentes aspectos de la estructura subyacente de los datos. Esto puede llevar a una partición más precisa que refleje mejor la complejidad de los datos.

- Mejora la estabilidad: crea agrupaciones de soluciones con menor sensibilidad al ruido y a los valores atípicos.

- Reducción del sesgo del algoritmo: cada algoritmo de agrupamiento tiene sus propias suposiciones y limitaciones. Al combinar múltiples algoritmos, es posible poder mitigar el sesgo inherente a cualquier algoritmo individual, lo que puede conducir a una partición más imparcial y equilibrada de los datos.

- Cribado de los datos: selecciona y filtra los mejores n resultados utilizando como medida de calidad el coeficiente de silueta.

- Consenso: realiza un recuento puro de los n mejores resultados,

4.4. El algoritmo *FuzzyMultiCHCClust-DS*

A lo largo de esta sección se presentará la propuesta del algoritmo evolutivo *FuzzyMultiCHCClust-DS* para flujo de datos. La sección está estructurada en tres partes. En la sección 4.4.1, se abordará la recepción y pre-procesamiento de los datos. La sección 4.4.2 se centrará en el post-procesamiento de la colección de datos. Finalmente, en la sección 4.4.3 se muestra el funcionamiento del algoritmo *FuzzyMultiCHCClust-DS*.

4.4.1. Colección de datos y pre-procesamiento

Este proceso se encarga de capturar los datos y organizarlos en lotes (*batchs* en inglés) de tamaño predefinido para luego ser tratados en una etapa a posteriori. La manera de capturarlo es mediante un recopilador que está continuamente recolectando datos provenientes de diversas fuentes de información. Estos datos, son agrupados según el orden de llegada dentro de lotes para posteriormente ser enviados. En este proceso los datos capturados tienen diferentes formatos y estructura, por lo que es necesario aplicar un proceso de transformación y normalización para estandarizar la información. Concluido el proceso de normalización y transformación de los datos del lote, es necesario establecer relaciones entre los datos contenido en el lote.

Para ello aplicamos un pre-procesamiento para relacionar datos. En la relación de los datos, los algoritmos de agrupamiento juegan un papel crucial, ya que son los responsables de etiquetar y organizar los datos en grupos similares. Se puede utilizar cualquier algoritmo de agrupamiento, destacando entre los más utilizados:

- Clara.
- Fanny.
- K-means.
- Pam.
- MiniBatchKMeans.

Tras completar esta fase de pre-procesamiento, se genera una matriz con los resultados de los algoritmos de agrupamiento. Un ejemplo de la matriz de resultados se puede visualizar en la figura 4.11. Cada fila de esta matriz contiene el resultado de un algoritmo de agrupamiento. La matriz de resultado será enviada para su procesamiento.

4.4.2. Procesamiento de los datos

Concluido el proceso de recolección de datos y su pre-procesamiento, entramos de lleno en el procesado. Para ello partimos de la matriz de resultados donde los datos ya están etiquetados. Los datos de la matriz de resultados pueden presentar algunos de los problemas tradicionales que tienen los algoritmos de agrupamiento como es la presencia de valores atípicos que pueden generar ruido y distorsionar de manera significativa los resultados. Para resolver estas limitaciones y generar un agrupamiento de calidad se utiliza la lógica difusa dentro del algoritmo *CHCClust*. Los resultados obtenidos de calcular *CHCClust* son evaluados mediante la aplicación del coeficiente de silueta para finalmente almacenarlos en la última columna de la matriz de resultados. A la columna que contiene los resultados del coeficiente de silueta se le aplica una función de cribado y consenso. Como resultado se obtiene un vector con el mejor agrupamiento posible.

4.4.3. Funcionamiento del algoritmo *FuzzyMultiCHCClust-DS*

El algoritmo *FuzzyMultiCHCClust-DS* está preparado para dar respuesta a problemas de optimización en entornos complejos. Su funcionamiento se organiza en dos etapas distintas para garantizar un procesamiento efectivo y exhaustivo de la información. En la primera etapa, se realiza la crucial tarea de recopilar datos, que implica la adquisición y preparación de conjuntos de datos para el análisis. Además, durante esta fase inicial, se lleva a cabo un riguroso proceso de normalización y transformación, con el objetivo de asegurar la coherencia de los datos antes de continuar con el análisis.

La segunda etapa del algoritmo entra en acción una vez que se ha completado la etapa de pre-procesamiento. En esta fase, la información resultante del proceso anterior se somete a un análisis más profundo. Aquí, se resuelven los problemas de incertidumbre y se aplican técnicas de optimización, cribado y consenso para seleccionar la mejor respuesta al conjunto de datos de entrada. La figura 4.11 proporciona una representación visual de este proceso, mostrando la secuencia de pasos desde la recopilación inicial de datos hasta el análisis final de los mismos.

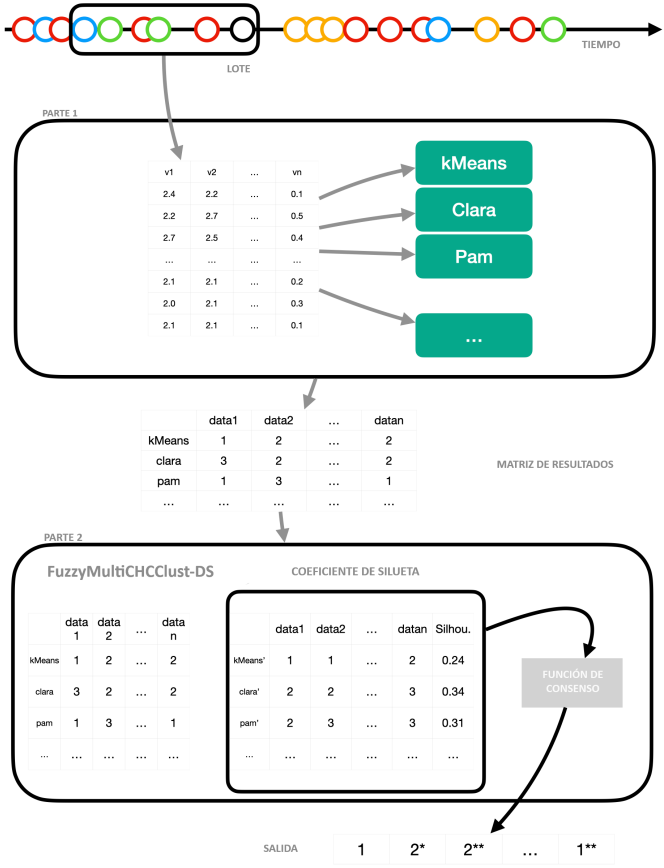


Figura 4.11: Esquema operacional del algoritmo *FuzzyMultiCHCClust-DS. Fuente: elaboración propia.*

La secuencia de instrucciones que compone el algoritmo *FuzzyMultiCHCClust-DS* se presenta en el algoritmo 4. En él se definen las entradas y salidas del algoritmo. Se desglosan las instrucciones que lo componen explicando la lógica que hay detrás de cada una de ellas. Terminada la explicación del pseudocódigo, debemos tener una visión integral y completa del funcionamiento del algoritmo, lo que facilitará su aplicación en diversas situaciones.

Algoritmo 4 Pseudocódigo del algoritmo *FuzzyMultiCHCClust-DS*.

```
Entrada
    lote: contiene la información de la matriz de resultados
    nMaxIteraciones: número máximo de iteraciones
    r_d: ratio divergencia
Salida
    resultado: vector con el mejor agrupamiento
1:  resultado ← []
2:  salida ← []
3:  for i from 1 to nrow(lote) do
4:      salida ← CHCClust(lote[i], nMaxIteraciones, r_d)
5:      lote[i][ncol(lote) + 1] ← CalcularCoeficienteSilueta(salida)
6:      salida ← []
7:  end for
8:  resultado ← CalcularConsenso(lote)
```

A continuación, se explica el funcionamiento:

1. *Líneas 1 -2*: inicializamos las variables.
2. *Líneas 4 - 6*: para cada una de las filas del `lote` aplicamos el algoritmo *CHCClust* y con el resultado obtenido le aplicamos el coeficiente de silueta. El resultado del coeficiente de silueta se almacena en la última columna del `lote`.
3. *Línea 8*: finalmente, los resultados del coeficiente de silueta serán consensuados y almacenados en un vector.

5
Estudio experimental

En esta sección se describe el estudio experimental realizado para la validación de las distintas propuestas presentadas en esta tesis. El objetivo es evaluar la eficacia y aplicabilidad de estas técnicas en contextos prácticos. Se analizarán diferentes conjuntos de datos para demostrar cómo optimizan los resultados las propuestas respecto a algoritmos ya existentes. Los resultados se analizan mediante test estadísticos que evalúan las hipótesis y determinan si las diferencias observadas en los resultados son significativas.

5.1. Diseño de la experimentación

En el ámbito de la investigación, es fundamental establecer un diseño experimental sólido que permita evaluar de manera precisa y significativa los algoritmos propuestos. Este diseño debe estar orientado a cumplir con los objetivos específicos del estudio. En este contexto, se usan técnicas avanzadas de validación, fundamentales para garantizar la robustez y la generalización de los resultados obtenidos.

Los algoritmos incluidos en este estudio serán evaluados sobre conjuntos de datos cuidadosamente seleccionados, que actúan como casos de prueba representativos de las situaciones reales. Estos conjuntos de datos se han obtenido de fuentes reconocidas como Kaggle y KEEL, con el objetivo de generar soluciones en diferentes contextos y condiciones.

Además, el diseño experimental incluye la aplicación de test estadísticos para analizar los resultados. Estos test nos permiten evaluar si hay evidencia suficiente en los datos para rechazar o no una hipótesis nula.

5.1.1. Conjuntos de datos

Para el estudio se han empleado 30 conjuntos de datos obtenidos de las páginas web de Kaggle [1] y KEEL [2]. Los conjuntos utilizados se detallan en la tabla 5.1. La tabla está compuesta de 4 columnas: la columna Nombre indica el nombre del conjunto de datos, la segunda columna representa el número de atributos que tiene el conjunto de datos, la tercera columna representa el número total de registros que compone el conjunto de datos y finalmente en la columna *Web* indicamos la web de la que se ha obtenido (Kaggle o KEEL).

[1] https://www.kaggle.com/
[2] https://sci2s.ugr.es/keel/datasets.php

Nombre	Atributos	Tamaño	Web
Alcohol's effect on young people	31	649	Kaggle
Basketball	5	96	KEEL
Bolts	8	40	KEEL
Credit Card Cheating Detection	31	284.807	Kaggle
College	19	777	Kaggle
ColorHistogram	33	68.040	KEEL
ColorMoments	10	68.040	KEEL
ColorTexture	17	68.040	KEEL
Country	10	166	Kaggle
Diabetes	10	768	Kaggle
Drug consumption	32	1.885	Kaggle
Fetal health	22	2.126	Kaggle
Haberman	3	306	KEEL
Heart	13	270	KEEL
Heart disease patients	12	303	Kaggle
House16H	17	22.784	KEEL
India leads report	31	23	Kaggle
Indicator districtwise health	9	281	Kaggle
Iris	4	150	KEEL
LayoutHistogram	33	66.616	KEEL
Parkinson	24	195	Kaggle
Pollution	16	60	KEEL
Price	10	950	KEEL
Quake	4	2.178	KEEL
Stulong	5	1.419	KEEL
Tae	5	151	KEEL
Transaction10K	3	120.427	KEEL
U.S.A. presidential results	21	3.141	Kaggle
Vehicle	16	159	Kaggle
Wine	13	178	KEEL
Wholesale customers	8	440	Kaggle

Tabla 5.1: Conjunto de datos utilizados en el estudio experimental.

5.1.2. Algoritmos

A continuación, se enumeran los algoritmos utilizados en el estudio experimental, junto con la configuración de los parámetros necesarios para su ejecución (ver tabla 5.2).

Algoritmo	Tipo agrupamiento	Parámetros
K-means	Particional	k = grupos iterMax = 10 metric = euclidean
Clara	Particional	k = grupos metric = euclidean
Pam	Particionamiento alrededor de los medoides	k = grupos variant = faster metric = euclidean
Fanny	Agrupamiento difuso	k = grupos iterMax = 500 metric = euclidean membExp = 2 tolOptimalInit = 1e-15
MiniBatchKmeans	Particional basado en lotes	k = grupos batchSize = 10 numInit = 100 iterMax = 100 initFraction = 1 earlyStopIter = 10 tolOptimalInit = 0.3
CHCClust	Algoritmo evolutivo	conjunto = datos n = 100
FuzzyMultiCHCClust-DS	Algoritmo evolutivo para entornos complejos	lote = 4096 nMaxIteraciones = 100 r_d = 0.35

Tabla 5.2: Algoritmos de agrupamiento y parámetros de configuración usados.

El significado de los distintos parámetros de configuración es el siguiente:

- *batchSize*: tamaño de los lotes.
- *conjunto*: representa el conjunto de datos.
- *earlyStopIter*: número de veces que debe ejecutarse después de calcular el mejor error cuadrático dentro del grupo.
- *initFraction*: porcentaje del conjunto de datos usado en la inicialización de los centroides.
- *iterMax*: número máximo permitido de iteraciones.
- *k*: representa el número de grupos en los que se divide el conjunto de datos.
- *lote*: conjunto de datos. Cada lote contiene los resultados con la matriz de resultados de los diferentes algoritmos de agrupamiento.
- *membExp*: exponente de pertenencia utilizado en el criterio de ajuste.
- *metric*: medida de distancia usada en el cálculo.
- *n*: número máximo de evaluaciones del algoritmo.
- *nMaxIteraciones*: número máximo de evaluaciones del algoritmo.
- *nEval*: número máximo de evaluaciones.
- *numInit*: número de veces que se ejecuta el algoritmo con diferentes centroides.
- r_d: porcentaje de bits que deben modificarse en el mejor cromosoma actual de la población para generar cada uno de los cromosomas restantes durante la reinicialización.
- *tolOptimalInit*: valor de tolerancia para el inicializador óptimo.
- *variant*: medida de distancia usada en el cálculo.

5.1.3. Test estadísticos

La estadística es un área de las matemáticas dedicada al tratamiento de los datos, que implica la recolección, organización y análisis. Su meta es entender cómo se comportan ciertos fenómenos haciendo uso de métodos numéricos. La estadística se organiza principalmente en dos ramas:

- Estadística descriptiva: formada por un grupo de métodos numéricos y visuales que explican y estudian un conjunto de datos, sin llegar a sacar conclusiones sobre el grupo más amplio al que pertenecen. Su objetivo es proporciona una visión general de un conjunto de datos con la finalidad de ayudar a identificar patrones, tendencias y características de los datos.

- Estadística inferencial: se define como el conjunto de enfoques estadísticos que facilitan la deducción de la distribución de la población y la inferencia de las relaciones entre variables utilizando los datos recolectados en la muestra. Por consiguiente, los principales propósitos de la inferencia estadística son estimar y evaluar hipótesis.

La estadística inferencial plantea el desafío de la toma de decisiones, donde tanto la estimación como las pruebas de hipótesis son elementos cruciales. Aunque son distintos entre sí, estos aspectos se complementan mutuamente. Los métodos paramétricos de la estadística inferencial se dividen principalmente en dos categorías:

- Métodos de estimación de parámetros: se emplean para calcular el valor de un parámetro desconocido de una población utilizando los datos de una muestra. Estas técnicas incluyen la estimación puntual, que ofrece un único valor como estimación del parámetro, y la estimación por intervalo, que proporciona un rango de valores en el que es probable que se encuentre el parámetro con cierto nivel de confianza.

- Métodos de prueba de hipótesis: se utilizan para evaluar afirmaciones sobre los parámetros de una población. Estos métodos implican formular una hipótesis nula y una hipótesis alternativa, recopilar datos y calcular una estadística de prueba a partir de la muestra. Luego, se compara la estadística de prueba con un valor crítico o se calcula un valor p para determinar si hay suficiente evidencia para rechazar la hipótesis nula en favor de la hipótesis alternativa. Estos métodos se dividen a su vez en dos grupos:

- Test paramétricos, que se caracterizan por tener una parametrización de dimensión infinita en las muestras donde se conoce el modelo de distribución de éstas. Para aplicar pruebas paramétricas, es esencial cumplir con ciertas condiciones; de lo contrario, el análisis estadístico carecería de credibilidad:
  - Uso de valores reales: la distinción principal entre pruebas paramétricas y no paramétricas radica en el nivel de medida representado por los datos. En este sentido, las pruebas paramétricas emplean datos compuestos por valores reales.
  - Independencia: es crucial que las muestras de estudio sean independientes entre sí, lo que implica que la ocurrencia o efecto en una población no debe influir en la probabilidad de ocurrencia o efecto en otras muestras de estudio.
  - Normalidad: este criterio se refiere al comportamiento de los datos. Para cumplir con esta condición, los datos deben seguir una distribución normal o gaussiana, con una media (μ) y una varianza (σ) específica.
- Test no paramétricos, no necesitan condiciones rigurosas para su aplicación en el análisis de datos; tampoco implica inferencias sobre los parámetros de la población ni requieren una parametrización de dimensión infinita. Además, no es preciso contar con datos que contengan valores reales; los datos pueden ser analizados en escalas nominal u ordinal.

Destacamos algunas de las distintas pruebas no paramétricas que son utilizadas para el análisis de los resultados:

- Test de *Friedman* [Friedman, 1937]: es una prueba estadística no paramétrica utilizada para comparar múltiples muestras relacionadas. Es especialmente útil cuando se quiere comparar más de dos grupos y no se cumplen las suposiciones necesarias para realizar un análisis de varianza de medidas repetidas (ANOVA de medidas repetidas).

  La hipótesis nula del Test de Friedman establece que no hay diferencias significativas entre los grupos. La hipótesis alternativa sugiere que al menos dos de los grupos difieren significativamente.

El cálculo de Friedman queda reflejado en la ecuación 5.1. El estadístico de Friedman sigue una distribución chi-cuadrado con $k-1$ grados de libertad, donde k es el número de grupos. Se compara el valor calculado del estadístico de Friedman con el valor crítico de la distribución chi-cuadrado para determinar si se rechaza la hipótesis nula.

$$\chi^2 = \frac{12}{Nk(k+1)} \left(\sum_{j=1}^{k} R_j^2 - \frac{k(k+1)^2}{4} \right) \tag{5.1}$$

donde

- χ^2 es el estadístico de *Friedman*.
- N es el número total de observaciones.
- k es el número de grupos.
- R_j es la suma de los rangos de las observaciones en el grupo j.

- Test de *Wilcoxon* [Wilcoxon, 1992]: llamado así en honor a Frank Wilcoxon quien lo publicó en 1945, es una prueba no paramétrica para comparar el rango medio de dos muestras relacionadas y determinar si existen diferencias significativas entre ellas. Al ser una prueba no paramétrica, no requiere suposiciones sobre la distribución de los datos. Se utiliza para evaluar si la diferencia entre dos mediciones relacionadas es significativa o si pudiera deberse al azar. Esta prueba es útil cuando se desea comparar dos conjuntos de datos relacionados, pero no se cumplen los supuestos de normalidad requeridos por pruebas paramétricas como la prueba t de Student.

$$W = \sum_{i=1}^{n} R_i \tag{5.2}$$

donde:

- n es el número total de pares de observaciones.
- R_i es el rango del i-ésimo par de observaciones.

Si tenemos los siguientes pares de observaciones:

$$(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)$$

La diferencia para el par (x_i, y_i) sería $|x_i - y_i|$. A continuación, estas diferencias son ordenadas de menor a mayor para asignar los rangos. El rango R_i para el par (x_i, y_i) sería el lugar que ocupa su diferencia en la secuencia ordenada de diferencias. Finalmente, se suman todos los R_i para obtener la estadística de prueba W.

- Test de *Holm* [Holm, 1979]: es un método de corrección para comparaciones múltiples, aplicable tanto en el contexto de test paramétricos como no paramétricos. La diferencia fundamental en su aplicación depende de la naturaleza de los datos y las suposiciones asociadas.

  Cuando se emplea el test de Holm en el marco del ranking de Friedman, su objetivo es identificar diferencias significativas entre el algoritmo de control (generalmente considerado como el mejor según el ranking de Friedman) y los demás algoritmos. Esto se logra al rechazar la hipótesis nula en favor del mejor algoritmo de clasificación.

  El test de Holm se utiliza como un procedimiento de corrección para comparaciones múltiples en test de hipótesis, ya sea en un contexto paramétrico o no paramétrico. Su aplicación específica en el ranking de Friedman busca determinar diferencias significativas entre un algoritmo de control y otros algoritmos bajo consideración.

  El método de corrección de Holm ajusta los valores p para comparaciones múltiples de la siguiente manera:

  1. Se ordenan los valores p de menor a mayor: $p_{(1)}, p_{(2)}, \ldots, p_{(m)}$, donde m es el número total de comparaciones.
  2. Para cada $i = 1, 2, \ldots, m$, se calcula $c_i = \frac{\alpha}{m-i+1}$, donde α es el nivel de significancia deseado.
  3. Se rechaza la hipótesis nula asociada con $p_{(i)}$ si $p_{(i)} < c_i$.

  La ecuación principal es $c_i = \frac{\alpha}{m-i+1}$, que determina el umbral ajustado para cada valor p ordenado. Si un valor p es menor que su correspondiente umbral c_i, entonces la hipótesis nula asociada con ese valor p se rechaza como significativa en el contexto de comparaciones múltiples.

5.2. Estudio experimental del algoritmo *CHCClust*

En primera instancia, es fundamental evaluar y analizar el desempeño del algoritmo *CHCClust* en comparación con los algoritmos tradicionales de agrupamiento, considerando además la cantidad de grupos utilizados para agrupar los datos. En la tabla 5.3 se muestran los resultados promedio de cada algoritmo, junto con el número de grupos obtenidos, utilizando como medida de calidad el coeficiente de silueta. Se resalta en negrita el mejor resultado promedio al comparar el agrupamiento original con *CHCClust*. Valores más cercanos a 1 indican una distribución óptima de los datos.

	Grupos				
	3	4	5	6	7
clara	0.1529	0.1187	0.1100	0.1000	0.0938
clara+*CHCClust*	**0.1690**⋆	**0.1318**⋆	**0.1225**	**0.1174**⋆	**0.1087**
fanny	0.1427	0.1174	0.1047	0.0920	0.1085
fanny+*CHCClust*	**0.1513**	**0.1253**	**0.1201**⋆	**0.1071**⋆	**0.1165**
kmeans	0.1554	0.1164	0.1116	0.0991	0.1047
kmeans+*CHCClust*	**0.1742**⋆	**0.1348**⋆	**0.1278**⋆	**0.1094**	**0.1067**
pam	0.1549	0.1212	0.1132	0.1047	0.0875
pam+*CHCClust*	**0.1771**⋆	**0.1352**⋆	**0.1276**⋆	**0.1257**⋆	**0.1051**⋆
minibatch	0.1424	0.1034	0.1194	0.0995	0.0974
minibatch+*CHCClust*	**0.1813**⋆	**0.1469**⋆	**0.1411**⋆	**0.1338**⋆	**0.1231**

Tabla 5.3: Resultado de la comparación por pares entre los algoritmos tradicionales y *CHCClust* para diversos números de grupos. Se indican con ⋆ las diferencias significativas que llevan al rechazo de la hipótesis de igualdad. Wilcoxon con $p < 0.05$

Los resultados completos obtenidos de la comparativa de *CHCClust* con algoritmos tradicionales están disponibles en el apéndice B.1.

5.2.1. Conclusión

El análisis de los resultados obtenidos por el algoritmo *CHCClust* ofrece las siguientes conclusiones:

- En todos los casos y para todos los números de grupos evaluados (de 3 a 7), los resultados mejoran significativamente cuando se utiliza *CHCClust* sobre los resultados de los algoritmos tradicionales (clara, fanny, k-means, pam y miniBatchKmeans).

- En términos de coeficiente de silueta, *CHCClust* muestra valores más altos en comparación con los algoritmos tradicionales en la mayoría de los casos. Las mejoras son especialmente notables en configuraciones con un mayor número de grupos (4, 5, 6, 7), donde las diferencias son estadísticamente significativas (indicadas por $\star$).

- *CHCClust* parece adaptarse bien a diferentes configuraciones de agrupamiento, demostrando mejoras significativas incluso cuando se combina con algoritmos que utilizan diversos tipos de agrupamiento como particionamiento, basados en medoides y basados en lotes.

En base a este análisis se puede afirmar que los resultados de esta tabla respaldan la efectividad de *CHCClust* como un algoritmo que mejora significativamente la calidad de los agrupamientos.

5.3. Estudio experimental del algoritmo *MultiCHCClust*

En este apartado se lleva a cabo un estudio para analizar y comparar el grado de cohesión y compactación de los grupos creados por algoritmos tradicionales de agrupamientos y el algoritmo *MultiCHCClust*. *MultiCHCClust* es un algoritmo post-procesamiento que se aplica sobre los resultados del algoritmo *CHCClust* para filtrar y seleccionar el mejor resultado de los posibles. Para ello hemos seleccionamos los resultados obtenidos por los diferentes algoritmos para $k = 3$, con el propósito de hacer un poco más simple el análisis y la evaluación de los resultados.

Lo que pretendemos demostrar es que el algoritmo *MultiCHCClust* es un algoritmo de post-procesamiento que mejora la calidad de los agrupamientos. En la tabla 5.4 tenemos los resultados medios obtenidos de la evaluación del coeficiente de silueta, y la diferencia significativa obtenida mediante el test de Holm.

	Algoritmos					
	Clara	Fanny	Kmeans	Pam	Minibatch	*MultiCHCClust*
Media	0.1529⋆	0.1427⋆	0.1554⋆	0.1549⋆	0.1386⋆	**0.1954**

Tabla 5.4: Resultados del cálculo medio del coeficiente de silueta para 3 grupos. Los campos marcados ⋆ presentan diferencias significativas del test de Holm con valor de $p < 0.01$.

Revisando los resultados podemos sacar en conclusión que la media del algoritmo *MultiCHCClust* mejora el agrupamiento en comparación con los algoritmos clara, fanny, k-means y pam. Además, se ha calculado el test de Holm para evaluar el rechazo de la hipótesis con $p < 0.01$. La tabla de resultados completa con el análisis de los algoritmos mencionados en la tabla 5.4 está disponible en el apéndice C.1.

Para completar el estudio, tenemos las tablas 5.6, 5.5 y 5.7 donde queda reflejado de forma visual el agrupamiento que han realizado cada uno de los algoritmos para el conjunto de datos de entrada. Las columnas de las tablas son marcadas con una serie de colores que representan el grupo donde el algoritmo ha ubicado los datos. Se ha realizado un agrupamiento para 3 grupos, donde el color verde indica que el dato ha sido englobado en el grupo 1, el color amarillo representa al grupo 2 y el color azul al grupo 3. Adicionalmente, se ha creado una fila con el resultado de la función de consenso. Los valores de esta fila son marcadas con los colores: rojo, naranja y blanco. Para el caso en el que la columna esté marcada en rojo, significa que no hay consenso, el naranja representa aquellos casos en los que hay un consenso parcial y el blanco indica que hay consenso completo. La tabla 5.6 muestra los resultados de la evaluación del conjunto de datos *Vehículo*, destacando que el consenso parcial y completo es mayoritario. De manera similar, en el conjunto de datos *Países* (tabla 5.7), se observa que el consenso parcial y completo predomina. Finalmente, para el conjunto de datos *Parkinson* (tabla 5.5), encontramos un consenso parcial casi en su totalidad.

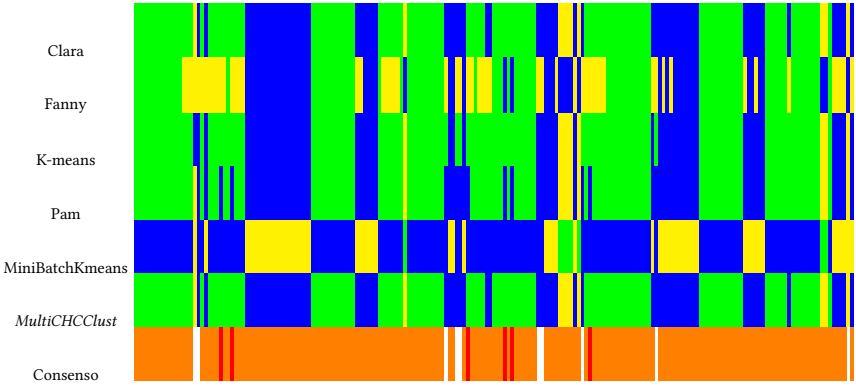


Tabla 5.5: Parkinson.

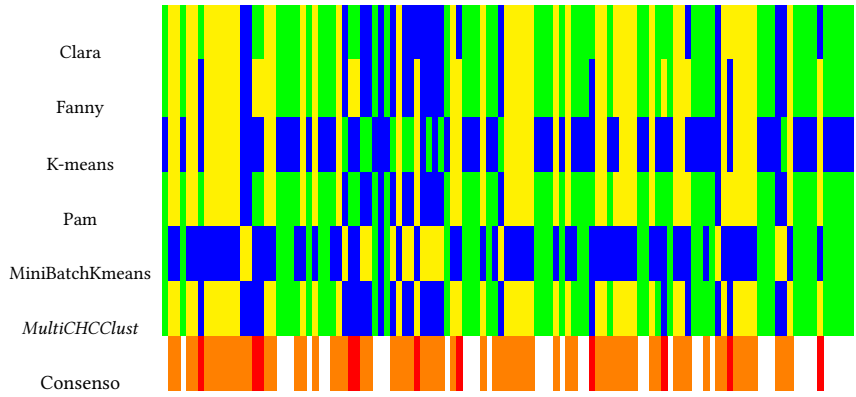


Tabla 5.6: Vehículo.

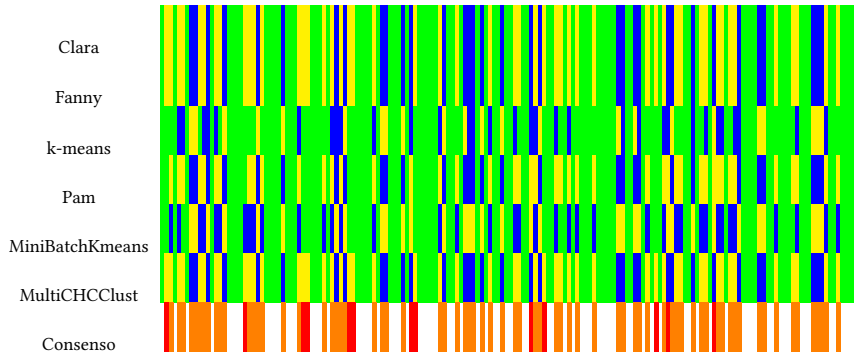


Tabla 5.7: Países.

5.3.1. Conclusión

Para cerrar esta sección, se ha evaluado la reacción del algoritmo *MultiCHCClust* frente a diferentes situaciones, destacando lo siguiente:

- Primero, hay que indicar que el algoritmo *MultiCHCClust* ha sido evaluado sobre un conjunto de casos de estudio, con la meta de analizar la capacidad del algoritmo para agrupar los datos en entornos complejos donde los datos pueden ser agrupados desde diferentes perspectivas.

- Segundo, los resultados del algoritmo *MultiCHCClust* han sido comparados con algoritmos de agrupamiento usados de forma tradicional para resolver problemas de agrupamiento. Estos resultados han sido evaluados con el test de Holm, demostrando que el algoritmo *MultiCHCClust* ofrece un desempeño notablemente mejor en comparación con el resto de los algoritmos. Hay que destacar, la representación gráfica para distribuir los datos del conjunto de datos en grupos, así como la respuesta del algoritmo *MultiCHCClust* para consensuar el resultado en los casos de divergencia.

- En resumen, *MultiCHCClust* nos ofrece resultados prometedores en la optimización de la distribución interna de un conjunto de datos para agrupamiento, además de su capacidad para consensuar los resultados, asegurando siempre la mejor estructura posible.

5.4. Estudio experimental del algoritmo *FuzzyMultiCHCClust-DS*

En esta sección, llevaremos a cabo un estudio experimental centrado en el algoritmo *FuzzyMultiCHCClust-DS*. Este algoritmo, basado en lógica difusa y el algoritmo *CHCClust*, representa una herramienta poderosa para el análisis de datos complejos y heterogéneos. Nuestro objetivo es examinar exhaustivamente su desempeño y su capacidad para abordar desafíos específicos en el proceso de agrupamiento de datos en entornos complejos. A través de este análisis experimental, abordaremos su eficacia en la extracción de conocimiento. Este estudio nos ayudará a entender mejor las capacidades del algoritmo *FuzzyMultiCHCClust-DS*, lo que contribuirá al progreso en el campo de la minería de flujos de datos continuos.

5.4.1. Conjunto de datos

El estudio se ha realizado empleando un conjunto de datos denominado Credit Card Cheating Detection, disponible en Kaggle [3]. Este conjunto de datos incluye transacciones efectuadas en septiembre de 2013 por titulares de tarjetas en Europa, entre las cuales se registraron 492 fraudes de un total de 284.807 transacciones. La clase positiva (fraudes) constituye el 0,172 % del total de transacciones, lo que evidencia un notable balance negativo en el conjunto de datos. Contiene 31 atributos, de los cuales dos son enteros y el resto decimales. Se descartaron los atributos `Time` y `Class` debido a su pobre influencia en el proceso de agrupamiento. Tras la exclusión de los atributos indicados en el análisis, el conjunto resultante contiene datos no etiquetados.

Para llevar a cabo el estudio experimental con el conjunto de datos, se realizó una simulación dividiendo el conjunto de datos en 69 lotes de 4096 registros y un lote de 2187 registros. El recolector recibió los lotes para enviarlos al algoritmo *FuzzyMultiCHCClust-DS*. La tabla 5.8 muestra algunas instancias del conjunto de datos utilizado en el estudio.

El significado de las columnas de la tabla 5.8 es el siguiente:

- $\#V_i$: resultado de una reducción de la dimensionalidad PCA para proteger las identidades de los usuarios y las características sensibles ($V_1 - V_{28}$).
- Amount: importe de la operación.

[3] https://www.kaggle.com/datasets/mlg-ulb/creditcardfraud

#V1	#V2	#V3	#V4	#V5	#V6	#V7	#V8	#V9		#V28	Amount
-1.35980	-0.07278	2.53634	1.37815	-0.33832	0.46238	0.23959	0.09869	0.36378	...	-0.02105	149.62
1.19185	0.26615	0.16648	0.44815	0.06001	-0.08236	-0.07880	0.08510	-0.25542	...	0.01472	2.69
-1.35835	-1.34016	1.77320	0.37977	-0.50319	1.80049	0.79146	0.24767	-1.51465	...	-0.05975	378.66
-0.96627	-0.18522	1.79299	-0.86329	-0.01030	1.24720	0.23760	0.37743	-1.38702	...	0.06145	123.5
-1.15823	0.87773	1.54871	0.40303	-0.40719	0.09592	0.59294	-0.27053	0.81773	...	0.21515	69.99
-0.42596	0.96052	1.14110	-0.16825	0.42098	-0.02972	0.47620	0.26031	-0.56867	...	0.08108	3.67

Tabla 5.8: Conjunto de datos de transacciones de titulares de tarjetas bancarias.

En la siguiente sección, nos centramos en la evaluación de los resultados del algoritmo *FuzzyMultiCHCClust-DS* utilizando este conjunto de datos.

5.4.2. Resultado

En esta sección se van a exponer los resultados obtenidos por el algoritmo *FuzzyMultiCHCClust-DS* para el conjunto de datos Credit Card Cheating Detection. Este algoritmo utiliza una variable lingüística con tres etiquetas en el proceso de agrupamiento. La evaluación de la calidad de los grupos generados se ha realizado utilizando la medida interna el coeficiente de silueta.

En la figura 5.1, se presentan los valores del coeficiente de silueta para los 70 lotes que han sido ejecutados por el algoritmo *FuzzyMultiCHCClust-DS*.

Las conclusiones que podemos extraer de la figura 5.1 son las siguientes:

- Variabilidad en la calidad de los grupos: los valores oscilan entre 0.47 y 0.60. Esto indica que los grupos tienen una buena compactación y separación interna de los datos.
- Implicaciones para la aplicación práctica: los resultados proporcionan una base sólida para utilizar el algoritmo *FuzzyMultiCHCClust-DS* en cualquier entorno donde se trabaje con datos complejos. La capacidad del algoritmo para generar grupos con buena calidad de agrupamiento, sugiere su utilidad.

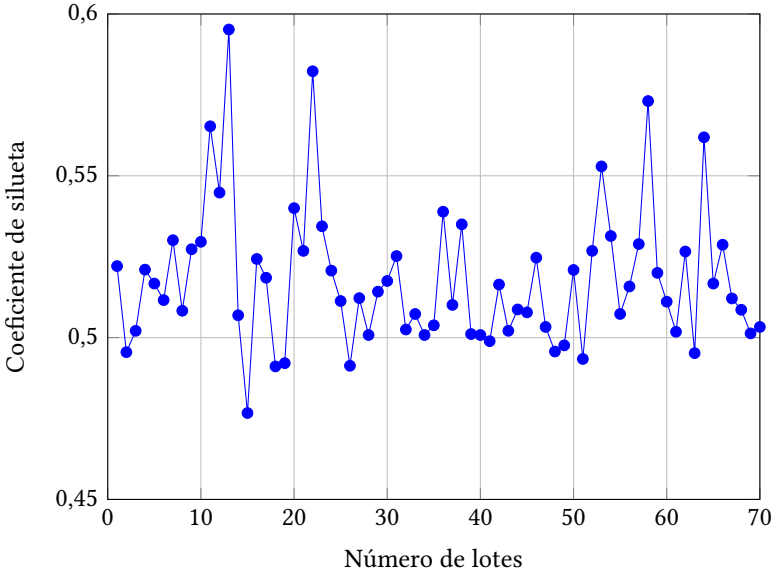


Figura 5.1: Coeficiente de silueta para los 70 lotes utilizando una variable lingüística con 3 etiquetas. *Fuente: elaboración propia.*

A Apéndice

A.1. Estudio experimental sobre la librería *Clustering*

En esta sección presentamos un estudio experimental utilizando la librería *Clustering*. El método *Clustering::clustering()* será ejecutado con la siguiente configuración:

- *path*: */Users/luis/Desktop/conjunto-datos/*. Directorio donde se ubican los conjuntos de datos de Kaggle [1].
- *min*: el mínimo número de grupos utilizado es 3.
- *max*: el máximo número de grupos utilizado es 4.
- *algorithm*: los algoritmos usados son: *Pam*, *Fanny* y *Clara*.
- *metrics*: las medidas utilizadas son: *Precision*, *Recall*, *Dunn* y *Silhouette*.

Las tablas A.1, A.2, A.3, A.4, A.5, A.6, A.7, A.8 y A.9 visualizan los resultados obtenidos de la ejecución del método *Clustering::clustering()* con la configuración indicada.

[1] https://www.kaggle.com/datasets

Apéndice

Algorithm	Distance	Clusters	Data	Var	Time	Precision	Recall
pam	euclidean	3	diabetes.csv	1^{st}	0.0702	0.8958	0.5716
pam	euclidean	3	diabetes.csv	2^{nd}	0.0708	0.2485	0.5161
pam	euclidean	3	diabetes.csv	3^{rd}	0.0709	0.1498	0.4673
pam	euclidean	3	diabetes.csv	4^{th}	0.0716	0.1277	0.4253
pam	euclidean	3	diabetes.csv	5^{th}	0.0735	0.0535	0.4108
pam	euclidean	3	diabetes.csv	6^{th}	0.0740	0.0455	0.4099
pam	euclidean	3	diabetes.csv	7^{th}	0.0759	0.0113	0.3707
pam	euclidean	3	diabetes.csv	8^{th}	0.0863	0.0060	0.3623
pam	euclidean	3	diabetes.csv	9^{th}	0.0878	0.0012	0.3387
pam	euclidean	3	diabetes.csv	10^{th}	0.3024	0.0000	0.0000
pam	euclidean	3	preeclampsia.csv	1^{st}	0.0165	0.5544	0.3883
pam	euclidean	3	preeclampsia.csv	2^{nd}	0.0166	0.5375	0.3770
pam	euclidean	3	preeclampsia.csv	3^{rd}	0.0169	0.5375	0.3687
pam	euclidean	3	preeclampsia.csv	4^{th}	0.0170	0.5326	0.3655
pam	euclidean	3	preeclampsia.csv	5^{th}	0.0170	0.5220	0.3646
pam	euclidean	3	preeclampsia.csv	6^{th}	0.0171	0.3798	0.3633
pam	euclidean	3	preeclampsia.csv	7^{th}	0.0172	0.3619	0.3599
pam	euclidean	3	preeclampsia.csv	8^{th}	0.0173	0.3509	0.3590
pam	euclidean	3	preeclampsia.csv	9^{th}	0.0175	0.3325	0.3566
pam	euclidean	3	preeclampsia.csv	10^{th}	0.0177	0.0579	0.3550
pam	euclidean	3	preeclampsia.csv	11^{th}	0.0178	0.0537	0.3550
pam	euclidean	3	preeclampsia.csv	12^{th}	0.0207	0.0530	0.3542
pam	euclidean	3	preeclampsia.csv	13^{th}	0.0224	0.0307	0.3452
pam	euclidean	3	preeclampsia.csv	14^{th}	0.0286	0.0157	0.3437
pam	euclidean	3	preeclampsia.csv	15^{th}	0.0294	0.0055	0.3387
pam	euclidean	3	preeclampsia.csv	16^{th}	0.0295	0.0000	0.0000
pam	euclidean	3	preeclampsia.csv	17^{th}	0.0297	0.0000	0.0000
pam	euclidean	3	preeclampsia.csv	18^{th}	0.0298	0.0000	0.0000
pam	euclidean	3	preeclampsia.csv	19^{th}	0.0298	0.0000	0.0000
pam	euclidean	3	preeclampsia.csv	20^{th}	0.0299	0.0000	0.0000
pam	euclidean	3	preeclampsia.csv	21^{st}	0.0301	0.0000	0.0000
pam	euclidean	3	preeclampsia.csv	22^{nd}	0.0302	0.0000	0.0000
pam	euclidean	3	preeclampsia.csv	23^{rd}	0.0303	0.0000	0.0000
pam	euclidean	3	preeclampsia.csv	24^{th}	0.0310	0.0000	0.0000
pam	euclidean	3	preeclampsia.csv	25^{th}	0.0344	0.0000	0.0000
pam	euclidean	4	diabetes.csv	1^{st}	0.0820	0.8854	0.4797
pam	euclidean	4	diabetes.csv	2^{nd}	0.0823	0.2825	0.4501
pam	euclidean	4	diabetes.csv	3^{rd}	0.0831	0.1539	0.4298
pam	euclidean	4	diabetes.csv	4^{th}	0.0842	0.1329	0.3757
pam	euclidean	4	diabetes.csv	5^{th}	0.0861	0.0580	0.3731
pam	euclidean	4	diabetes.csv	6^{th}	0.0888	0.0459	0.3497
pam	euclidean	4	diabetes.csv	7^{th}	0.0893	0.0121	0.3483
pam	euclidean	4	diabetes.csv	8^{th}	0.0993	0.0060	0.3176
pam	euclidean	4	diabetes.csv	9^{th}	0.0994	0.0013	0.2930
pam	euclidean	4	diabetes.csv	10^{th}	0.1118	0.0000	0.0000
pam	euclidean	4	preeclampsia.csv	1^{st}	0.0188	0.5654	0.2874

Tabla A.1: Resultados obtenidos por la librería *Clustering* para el algoritmo *Pam*.

Algorithm	Distance	Clusters	Data	Var	Time	Precision	Recall
pam	euclidean	4	preeclampsia.csv	2^{nd}	0.0190	0.5623	0.2867
pam	euclidean	4	preeclampsia.csv	3^{rd}	0.0191	0.5588	0.2864
pam	euclidean	4	preeclampsia.csv	4^{th}	0.0191	0.5372	0.2849
pam	euclidean	4	preeclampsia.csv	5^{th}	0.0191	0.5279	0.2818
pam	euclidean	4	preeclampsia.csv	6^{th}	0.0193	0.3749	0.2795
pam	euclidean	4	preeclampsia.csv	7^{th}	0.0193	0.3645	0.2784
pam	euclidean	4	preeclampsia.csv	8^{th}	0.0198	0.3507	0.2732
pam	euclidean	4	preeclampsia.csv	9^{th}	0.0199	0.3383	0.2711
pam	euclidean	4	preeclampsia.csv	10^{th}	0.0203	0.0633	0.2691
pam	euclidean	4	preeclampsia.csv	11^{th}	0.0207	0.0534	0.2687
pam	euclidean	4	preeclampsia.csv	12^{th}	0.0215	0.0529	0.2659
pam	euclidean	4	preeclampsia.csv	13^{th}	0.0215	0.0301	0.2642
pam	euclidean	4	preeclampsia.csv	14^{th}	0.0254	0.0161	0.2601
pam	euclidean	4	preeclampsia.csv	15^{th}	0.0315	0.0057	0.2584
pam	euclidean	4	preeclampsia.csv	16^{th}	0.0324	0.0000	0.0000
pam	euclidean	4	preeclampsia.csv	17^{th}	0.0325	0.0000	0.0000
pam	euclidean	4	preeclampsia.csv	18^{th}	0.0326	0.0000	0.0000
pam	euclidean	4	preeclampsia.csv	19^{th}	0.0327	0.0000	0.0000
pam	euclidean	4	preeclampsia.csv	20^{th}	0.0329	0.0000	0.0000
pam	euclidean	4	preeclampsia.csv	21^{st}	0.0332	0.0000	0.0000
pam	euclidean	4	preeclampsia.csv	22^{nd}	0.0336	0.0000	0.0000
pam	euclidean	4	preeclampsia.csv	23^{rd}	0.0337	0.0000	0.0000
pam	euclidean	4	preeclampsia.csv	24^{th}	0.0338	0.0000	0.0000
pam	euclidean	4	preeclampsia.csv	25^{th}	0.0345	0.0000	0.0000
pam	manhattan	3	diabetes.csv	1^{st}	0.0875	0.8138	0.5719
pam	manhattan	3	diabetes.csv	2^{nd}	0.0884	0.2801	0.5161
pam	manhattan	3	diabetes.csv	3^{rd}	0.0896	0.1670	0.4787
pam	manhattan	3	diabetes.csv	4^{th}	0.0914	0.1316	0.4356
pam	manhattan	3	diabetes.csv	5^{th}	0.0919	0.0552	0.4254
pam	manhattan	3	diabetes.csv	6^{th}	0.0943	0.0475	0.4059
pam	manhattan	3	diabetes.csv	7^{th}	0.0959	0.0118	0.3945
pam	manhattan	3	diabetes.csv	8^{th}	0.1061	0.0058	0.3850
pam	manhattan	3	diabetes.csv	9^{th}	0.1089	0.0014	0.3710
pam	manhattan	3	diabetes.csv	10^{th}	0.1116	0.0000	0.0000
pam	manhattan	3	preeclampsia.csv	1^{st}	0.0159	0.6102	0.4100
pam	manhattan	3	preeclampsia.csv	2^{nd}	0.0161	0.5605	0.3824
pam	manhattan	3	preeclampsia.csv	3^{rd}	0.0161	0.5455	0.3775
pam	manhattan	3	preeclampsia.csv	4^{th}	0.0161	0.5415	0.3701
pam	manhattan	3	preeclampsia.csv	5^{th}	0.0162	0.5273	0.3679
pam	manhattan	3	preeclampsia.csv	6^{th}	0.0163	0.3777	0.3650
pam	manhattan	3	preeclampsia.csv	7^{th}	0.0163	0.3668	0.3647
pam	manhattan	3	preeclampsia.csv	8^{th}	0.0164	0.3370	0.3635
pam	manhattan	3	preeclampsia.csv	9^{th}	0.0165	0.3345	0.3551
pam	manhattan	3	preeclampsia.csv	10^{th}	0.0167	0.0576	0.3506
pam	manhattan	3	preeclampsia.csv	11^{th}	0.0170	0.0542	0.3440
pam	manhattan	3	preeclampsia.csv	12^{th}	0.0173	0.0523	0.3400

Tabla A.2: Resultados obtenidos por la librería *Clustering* para el algoritmo *Pam*.

Algorithm	Distance	Clusters	Data	Var	Time	Precision	Recall
pam	manhattan	3	preeclampsia.csv	13^{th}	0.0195	0.0301	0.3387
pam	manhattan	3	preeclampsia.csv	14^{th}	0.0234	0.0169	0.3351
pam	manhattan	3	preeclampsia.csv	15^{th}	0.0286	0.0046	0.2988
pam	manhattan	3	preeclampsia.csv	16^{th}	0.0290	0.0000	0.0000
pam	manhattan	3	preeclampsia.csv	17^{th}	0.0290	0.0000	0.0000
pam	manhattan	3	preeclampsia.csv	18^{th}	0.0292	0.0000	0.0000
pam	manhattan	3	preeclampsia.csv	19^{th}	0.0292	0.0000	0.0000
pam	manhattan	3	preeclampsia.csv	20^{th}	0.0293	0.0000	0.0000
pam	manhattan	3	preeclampsia.csv	21^{st}	0.0297	0.0000	0.0000
pam	manhattan	3	preeclampsia.csv	22^{nd}	0.0301	0.0000	0.0000
pam	manhattan	3	preeclampsia.csv	23^{rd}	0.0312	0.0000	0.0000
pam	manhattan	3	preeclampsia.csv	24^{th}	0.0323	0.0000	0.0000
pam	manhattan	3	preeclampsia.csv	25^{th}	0.0406	0.0000	0.0000
pam	manhattan	4	diabetes.csv	1^{st}	0.0561	0.8568	0.4865
pam	manhattan	4	diabetes.csv	2^{nd}	0.0577	0.2678	0.4656
pam	manhattan	4	diabetes.csv	3^{rd}	0.0577	0.1658	0.4559
pam	manhattan	4	diabetes.csv	4^{th}	0.0587	0.1433	0.4064
pam	manhattan	4	diabetes.csv	5^{th}	0.0621	0.0613	0.3763
pam	manhattan	4	diabetes.csv	6^{th}	0.0629	0.0480	0.3342
pam	manhattan	4	diabetes.csv	7^{th}	0.0662	0.0121	0.3330
pam	manhattan	4	diabetes.csv	8^{th}	0.0727	0.0057	0.3326
pam	manhattan	4	diabetes.csv	9^{th}	0.0735	0.0014	0.3226
pam	manhattan	4	diabetes.csv	10^{th}	0.0843	0.0000	0.0000
pam	manhattan	4	preeclampsia.csv	1^{st}	0.0179	0.5824	0.2931
pam	manhattan	4	preeclampsia.csv	2^{nd}	0.0180	0.5781	0.2909
pam	manhattan	4	preeclampsia.csv	3^{rd}	0.0182	0.5721	0.2898
pam	manhattan	4	preeclampsia.csv	4^{th}	0.0182	0.5689	0.2886
pam	manhattan	4	preeclampsia.csv	5^{th}	0.0183	0.5386	0.2870
pam	manhattan	4	preeclampsia.csv	6^{th}	0.0184	0.3835	0.2869
pam	manhattan	4	preeclampsia.csv	7^{th}	0.0184	0.3657	0.2786
pam	manhattan	4	preeclampsia.csv	8^{th}	0.0185	0.3464	0.2764
pam	manhattan	4	preeclampsia.csv	9^{th}	0.0189	0.3402	0.2754
pam	manhattan	4	preeclampsia.csv	10^{th}	0.0191	0.0583	0.2732
pam	manhattan	4	preeclampsia.csv	11^{th}	0.0191	0.0555	0.2717
pam	manhattan	4	preeclampsia.csv	12^{th}	0.0197	0.0543	0.2600
pam	manhattan	4	preeclampsia.csv	13^{th}	0.0207	0.0322	0.2571
pam	manhattan	4	preeclampsia.csv	14^{th}	0.0213	0.0176	0.2568
pam	manhattan	4	preeclampsia.csv	15^{th}	0.0253	0.0051	0.2458
pam	manhattan	4	preeclampsia.csv	16^{th}	0.0313	0.0000	0.0000
pam	manhattan	4	preeclampsia.csv	17^{th}	0.0316	0.0000	0.0000
pam	manhattan	4	preeclampsia.csv	18^{th}	0.0316	0.0000	0.0000
pam	manhattan	4	preeclampsia.csv	19^{th}	0.0320	0.0000	0.0000
pam	manhattan	4	preeclampsia.csv	20^{th}	0.0320	0.0000	0.0000
pam	manhattan	4	preeclampsia.csv	21^{st}	0.0321	0.0000	0.0000
pam	manhattan	4	preeclampsia.csv	22^{nd}	0.0322	0.0000	0.0000
pam	manhattan	4	preeclampsia.csv	23^{rd}	0.0327	0.0000	0.0000

Tabla A.3: Resultados obtenidos por la librería *Clustering* para el algoritmo *Pam*.

Algorithm	Distance	Clusters	Data	Var	Time	Precision	Recall
clara	euclidean	3	diabetes.csv	1^{st}	0.0050	0.7801	0.7183
clara	euclidean	3	diabetes.csv	2^{nd}	0.0057	0.2381	0.6205
clara	euclidean	3	diabetes.csv	3^{rd}	0.0058	0.1396	0.5797
clara	euclidean	3	diabetes.csv	4^{th}	0.0061	0.1294	0.5559
clara	euclidean	3	diabetes.csv	5^{th}	0.0088	0.0660	0.5224
clara	euclidean	3	diabetes.csv	6^{th}	0.0106	0.0458	0.4973
clara	euclidean	3	diabetes.csv	7^{th}	0.0135	0.0115	0.4654
clara	euclidean	3	diabetes.csv	8^{th}	0.0218	0.0053	0.4501
clara	euclidean	3	diabetes.csv	9^{th}	0.0231	0.0014	0.4327
clara	euclidean	3	diabetes.csv	10^{th}	0.0441	0.0000	0.0000
clara	euclidean	3	preeclampsia.csv	1^{st}	0.0036	0.5645	0.3876
clara	euclidean	3	preeclampsia.csv	2^{nd}	0.0039	0.5398	0.3855
clara	euclidean	3	preeclampsia.csv	3^{rd}	0.0039	0.5338	0.3830
clara	euclidean	3	preeclampsia.csv	4^{th}	0.0039	0.5279	0.3766
clara	euclidean	3	preeclampsia.csv	5^{th}	0.0041	0.5114	0.3686
clara	euclidean	3	preeclampsia.csv	6^{th}	0.0041	0.3776	0.3644
clara	euclidean	3	preeclampsia.csv	7^{th}	0.0043	0.3501	0.3600
clara	euclidean	3	preeclampsia.csv	8^{th}	0.0043	0.3433	0.3596
clara	euclidean	3	preeclampsia.csv	9^{th}	0.0043	0.3408	0.3582
clara	euclidean	3	preeclampsia.csv	10^{th}	0.0046	0.0569	0.3566
clara	euclidean	3	preeclampsia.csv	11^{th}	0.0047	0.0554	0.3512
clara	euclidean	3	preeclampsia.csv	12^{th}	0.0048	0.0529	0.3491
clara	euclidean	3	preeclampsia.csv	13^{th}	0.0056	0.0291	0.3462
clara	euclidean	3	preeclampsia.csv	14^{th}	0.0063	0.0162	0.3396
clara	euclidean	3	preeclampsia.csv	15^{th}	0.0099	0.0059	0.3372
clara	euclidean	3	preeclampsia.csv	16^{th}	0.0166	0.0000	0.0000
clara	euclidean	3	preeclampsia.csv	17^{th}	0.0166	0.0000	0.0000
clara	euclidean	3	preeclampsia.csv	18^{th}	0.0168	0.0000	0.0000
clara	euclidean	3	preeclampsia.csv	19^{th}	0.0168	0.0000	0.0000
clara	euclidean	3	preeclampsia.csv	20^{th}	0.0170	0.0000	0.0000
clara	euclidean	3	preeclampsia.csv	21^{st}	0.0170	0.0000	0.0000
clara	euclidean	3	preeclampsia.csv	22^{nd}	0.0171	0.0000	0.0000
clara	euclidean	3	preeclampsia.csv	23^{rd}	0.0172	0.0000	0.0000
clara	euclidean	3	preeclampsia.csv	24^{th}	0.0174	0.0000	0.0000
clara	euclidean	3	preeclampsia.csv	25^{th}	0.0179	0.0000	0.0000
clara	euclidean	4	diabetes.csv	1^{st}	0.0051	0.7783	0.4423
clara	euclidean	4	diabetes.csv	2^{nd}	0.0059	0.2978	0.3854
clara	euclidean	4	diabetes.csv	3^{rd}	0.0071	0.1687	0.3779
clara	euclidean	4	diabetes.csv	4^{th}	0.0071	0.1292	0.3451
clara	euclidean	4	diabetes.csv	5^{th}	0.0080	0.0580	0.3369
clara	euclidean	4	diabetes.csv	6^{th}	0.0099	0.0480	0.3305
clara	euclidean	4	diabetes.csv	7^{th}	0.0151	0.0125	0.3274
clara	euclidean	4	diabetes.csv	8^{th}	0.0168	0.0065	0.3145
clara	euclidean	4	diabetes.csv	9^{th}	0.0273	0.0015	0.2979
clara	euclidean	4	diabetes.csv	10^{th}	0.0304	0.0000	0.0000
clara	euclidean	4	preeclampsia.csv	1^{st}	0.0043	0.5609	0.3108

Tabla A.4: Resultados obtenidos por la librería *Clustering* para el algoritmo *Clara*.

Algorithm	Distance	Clusters	Data	Var	Time	Precision	Recall
clara	euclidean	4	preeclampsia.csv	2^{nd}	0.0043	0.5473	0.3040
clara	euclidean	4	preeclampsia.csv	3^{rd}	0.0044	0.5457	0.3028
clara	euclidean	4	preeclampsia.csv	4^{th}	0.0044	0.5448	0.3001
clara	euclidean	4	preeclampsia.csv	5^{th}	0.0045	0.5350	0.2997
clara	euclidean	4	preeclampsia.csv	6^{th}	0.0045	0.3659	0.2985
clara	euclidean	4	preeclampsia.csv	7^{th}	0.0046	0.3615	0.2972
clara	euclidean	4	preeclampsia.csv	8^{th}	0.0049	0.3388	0.2969
clara	euclidean	4	preeclampsia.csv	9^{th}	0.0051	0.3373	0.2933
clara	euclidean	4	preeclampsia.csv	10^{th}	0.0051	0.0591	0.2918
clara	euclidean	4	preeclampsia.csv	11^{th}	0.0053	0.0558	0.2854
clara	euclidean	4	preeclampsia.csv	12^{th}	0.0059	0.0525	0.2828
clara	euclidean	4	preeclampsia.csv	13^{th}	0.0060	0.0324	0.2818
clara	euclidean	4	preeclampsia.csv	14^{th}	0.0067	0.0161	0.2768
clara	euclidean	4	preeclampsia.csv	15^{th}	0.0102	0.0059	0.2756
clara	euclidean	4	preeclampsia.csv	16^{th}	0.0176	0.0000	0.0000
clara	euclidean	4	preeclampsia.csv	17^{th}	0.0177	0.0000	0.0000
clara	euclidean	4	preeclampsia.csv	18^{th}	0.0178	0.0000	0.0000
clara	euclidean	4	preeclampsia.csv	19^{th}	0.0183	0.0000	0.0000
clara	euclidean	4	preeclampsia.csv	20^{th}	0.0185	0.0000	0.0000
clara	euclidean	4	preeclampsia.csv	21^{st}	0.0185	0.0000	0.0000
clara	euclidean	4	preeclampsia.csv	22^{nd}	0.0186	0.0000	0.0000
clara	euclidean	4	preeclampsia.csv	23^{rd}	0.0188	0.0000	0.0000
clara	euclidean	4	preeclampsia.csv	24^{th}	0.0188	0.0000	0.0000
clara	euclidean	4	preeclampsia.csv	25^{th}	0.0224	0.0000	0.0000
clara	manhattan	3	diabetes.csv	1^{st}	0.0048	0.7771	0.6955
clara	manhattan	3	diabetes.csv	2^{nd}	0.0056	0.2519	0.6137
clara	manhattan	3	diabetes.csv	3^{rd}	0.0060	0.1410	0.6012
clara	manhattan	3	diabetes.csv	4^{th}	0.0067	0.1377	0.5674
clara	manhattan	3	diabetes.csv	5^{th}	0.0067	0.0643	0.5163
clara	manhattan	3	diabetes.csv	6^{th}	0.0096	0.0476	0.4812
clara	manhattan	3	diabetes.csv	7^{th}	0.0106	0.0115	0.4802
clara	manhattan	3	diabetes.csv	8^{th}	0.0130	0.0053	0.4547
clara	manhattan	3	diabetes.csv	9^{th}	0.0223	0.0014	0.4501
clara	manhattan	3	diabetes.csv	10^{th}	0.0286	0.0000	0.0000
clara	manhattan	3	preeclampsia.csv	1^{st}	0.0038	0.5632	0.3902
clara	manhattan	3	preeclampsia.csv	2^{nd}	0.0039	0.5441	0.3849
clara	manhattan	3	preeclampsia.csv	3^{rd}	0.0040	0.5329	0.3824
clara	manhattan	3	preeclampsia.csv	4^{th}	0.0042	0.5259	0.3819
clara	manhattan	3	preeclampsia.csv	5^{th}	0.0043	0.5249	0.3794
clara	manhattan	3	preeclampsia.csv	6^{th}	0.0046	0.3711	0.3783
clara	manhattan	3	preeclampsia.csv	7^{th}	0.0046	0.3656	0.3707
clara	manhattan	3	preeclampsia.csv	8^{th}	0.0049	0.3648	0.3695
clara	manhattan	3	preeclampsia.csv	9^{th}	0.0050	0.3366	0.3672
clara	manhattan	3	preeclampsia.csv	10^{th}	0.0050	0.0603	0.3666
clara	manhattan	3	preeclampsia.csv	11^{th}	0.0055	0.0524	0.3659
clara	manhattan	3	preeclampsia.csv	12^{th}	0.0064	0.0502	0.3633

Tabla A.5: Resultados obtenidos por la librería *Clustering* para el algoritmo *Clara*.

Algorithm	Distance	Clusters	Data	Var	Time	Precision	Recall
clara	manhattan	3	preeclampsia.csv	13	0.0077	0.0302	0.3572
clara	manhattan	3	preeclampsia.csv	14	0.0095	0.0164	0.3516
clara	manhattan	3	preeclampsia.csv	15	0.0096	0.0057	0.3490
clara	manhattan	3	preeclampsia.csv	16^{th}	0.0165	0.0000	0.0000
clara	manhattan	3	preeclampsia.csv	17^{th}	0.0169	0.0000	0.0000
clara	manhattan	3	preeclampsia.csv	18^{th}	0.0171	0.0000	0.0000
clara	manhattan	3	preeclampsia.csv	19^{th}	0.0173	0.0000	0.0000
clara	manhattan	3	preeclampsia.csv	20^{th}	0.0174	0.0000	0.0000
clara	manhattan	3	preeclampsia.csv	21^{st}	0.0175	0.0000	0.0000
clara	manhattan	3	preeclampsia.csv	22^{nd}	0.0182	0.0000	0.0000
clara	manhattan	3	preeclampsia.csv	23^{rd}	0.0191	0.0000	0.0000
clara	manhattan	3	preeclampsia.csv	24^{th}	0.0200	0.0000	0.0000
clara	manhattan	3	preeclampsia.csv	25^{th}	0.2075	0.0000	0.0000
clara	manhattan	4	diabetes.csv	1^{st}	0.0047	0.6663	0.6806
clara	manhattan	4	diabetes.csv	2^{nd}	0.0054	0.3757	0.4145
clara	manhattan	4	diabetes.csv	3^{rd}	0.0066	0.2611	0.4030
clara	manhattan	4	diabetes.csv	4^{th}	0.0067	0.1421	0.3580
clara	manhattan	4	diabetes.csv	5^{th}	0.0073	0.0610	0.3554
clara	manhattan	4	diabetes.csv	6^{th}	0.0099	0.0497	0.3217
clara	manhattan	4	diabetes.csv	7^{th}	0.0170	0.0129	0.3067
clara	manhattan	4	diabetes.csv	8^{th}	0.0208	0.0059	0.3061
clara	manhattan	4	diabetes.csv	9^{th}	0.0291	0.0014	0.2957
clara	manhattan	4	diabetes.csv	10^{th}	0.0309	0.0000	0.0000
clara	manhattan	4	preeclampsia.csv	1^{st}	0.0044	0.5804	0.3084
clara	manhattan	4	preeclampsia.csv	2^{nd}	0.0048	0.5712	0.3055
clara	manhattan	4	preeclampsia.csv	3^{rd}	0.0050	0.5580	0.3036
clara	manhattan	4	preeclampsia.csv	4^{th}	0.0053	0.5377	0.2984
clara	manhattan	4	preeclampsia.csv	5^{th}	0.0059	0.5274	0.2912
clara	manhattan	4	preeclampsia.csv	6^{th}	0.0061	0.3816	0.2909
clara	manhattan	4	preeclampsia.csv	7^{th}	0.0062	0.3622	0.2869
clara	manhattan	4	preeclampsia.csv	8^{th}	0.0066	0.3516	0.2849
clara	manhattan	4	preeclampsia.csv	9^{th}	0.0067	0.3402	0.2828
clara	manhattan	4	preeclampsia.csv	10^{th}	0.0073	0.0610	0.2820
clara	manhattan	4	preeclampsia.csv	11^{th}	0.0077	0.0531	0.2817
clara	manhattan	4	preeclampsia.csv	12^{th}	0.0103	0.0519	0.2767
clara	manhattan	4	preeclampsia.csv	13^{th}	0.0168	0.0295	0.2725
clara	manhattan	4	preeclampsia.csv	14^{th}	0.0169	0.0169	0.2699
clara	manhattan	4	preeclampsia.csv	15^{th}	0.0178	0.0053	0.2673
clara	manhattan	4	preeclampsia.csv	16^{th}	0.0182	0.0000	0.0000
clara	manhattan	4	preeclampsia.csv	17^{th}	0.0183	0.0000	0.0000
clara	manhattan	4	preeclampsia.csv	18^{th}	0.0185	0.0000	0.0000
clara	manhattan	4	preeclampsia.csv	19^{th}	0.0189	0.0000	0.0000
clara	manhattan	4	preeclampsia.csv	20^{th}	0.0190	0.0000	0.0000
clara	manhattan	4	preeclampsia.csv	21^{st}	0.0192	0.0000	0.0000
clara	manhattan	4	preeclampsia.csv	22^{nd}	0.0196	0.0000	0.0000
clara	manhattan	4	preeclampsia.csv	23^{rd}	0.0210	0.0000	0.0000
clara	manhattan	4	preeclampsia.csv	24^{th}	0.0211	0.0000	0.0000
clara	manhattan	4	preeclampsia.csv	25^{th}	0.0215	0.0000	0.0000

Tabla A.6: Resultados obtenidos por la librería *Clustering* para el algoritmo *Clara*.

Algorithm	Distance	Clusters	Data	Var	Time	Precision	Recall
fanny	euclidean	3	diabetes.csv	1^{st}	0.2960	0.6677	0.8270
fanny	euclidean	3	diabetes.csv	2^{nd}	0.3002	0.2347	0.7899
fanny	euclidean	3	diabetes.csv	3^{rd}	0.3014	0.1235	0.7625
fanny	euclidean	3	diabetes.csv	4^{th}	0.3069	0.0972	0.7445
fanny	euclidean	3	diabetes.csv	5^{th}	0.3084	0.0511	0.6774
fanny	euclidean	3	diabetes.csv	6^{th}	0.3091	0.0436	0.6597
fanny	euclidean	3	diabetes.csv	7^{th}	0.3096	0.0110	0.6593
fanny	euclidean	3	diabetes.csv	8^{th}	0.3112	0.0052	0.6345
fanny	euclidean	3	diabetes.csv	9^{th}	0.3214	0.0013	0.6210
fanny	euclidean	3	diabetes.csv	10^{th}	0.3316	0.0000	0.0000
fanny	euclidean	3	preeclampsia.csv	1^{st}	0.0446	0.5020	0.7942
fanny	euclidean	3	preeclampsia.csv	2^{nd}	0.0449	0.5005	0.7837
fanny	euclidean	3	preeclampsia.csv	3^{rd}	0.0452	0.4996	0.7759
fanny	euclidean	3	preeclampsia.csv	4^{th}	0.0454	0.4992	0.7749
fanny	euclidean	3	preeclampsia.csv	5^{th}	0.0456	0.4984	0.7744
fanny	euclidean	3	preeclampsia.csv	6^{th}	0.0460	0.3360	0.7734
fanny	euclidean	3	preeclampsia.csv	7^{th}	0.0466	0.3326	0.7727
fanny	euclidean	3	preeclampsia.csv	8^{th}	0.0469	0.3324	0.7725
fanny	euclidean	3	preeclampsia.csv	9^{th}	0.0474	0.3315	0.7724
fanny	euclidean	3	preeclampsia.csv	10^{th}	0.0487	0.0570	0.7722
fanny	euclidean	3	preeclampsia.csv	11^{th}	0.0492	0.0500	0.7722
fanny	euclidean	3	preeclampsia.csv	12^{th}	0.0502	0.0497	0.7720
fanny	euclidean	3	preeclampsia.csv	13^{th}	0.0571	0.0298	0.7719
fanny	euclidean	3	preeclampsia.csv	14^{th}	0.0577	0.0159	0.7679
fanny	euclidean	3	preeclampsia.csv	15^{th}	0.0618	0.0051	0.7542
fanny	euclidean	3	preeclampsia.csv	16	0.0622	0.0000	0.0000
fanny	euclidean	3	preeclampsia.csv	17	0.0623	0.0000	0.0000
fanny	euclidean	3	preeclampsia.csv	18	0.0628	0.0000	0.0000
fanny	euclidean	3	preeclampsia.csv	19	0.0636	0.0000	0.0000
fanny	euclidean	3	preeclampsia.csv	20	0.0641	0.0000	0.0000
fanny	euclidean	3	preeclampsia.csv	21	0.0655	0.0000	0.0000
fanny	euclidean	3	preeclampsia.csv	22	0.0679	0.0000	0.0000
fanny	euclidean	3	preeclampsia.csv	23	0.0735	0.0000	0.0000
fanny	euclidean	3	preeclampsia.csv	24	0.0764	0.0000	0.0000
fanny	euclidean	3	preeclampsia.csv	25	0.1364	0.0000	0.0000
fanny	euclidean	4	diabetes.csv	1^{st}	0.4556	0.5820	0.8934
fanny	euclidean	4	diabetes.csv	2^{nd}	0.4603	0.2426	0.8901
fanny	euclidean	4	diabetes.csv	3^{rd}	0.4642	0.1114	0.8760
fanny	euclidean	4	diabetes.csv	4^{th}	0.4816	0.1012	0.8746
fanny	euclidean	4	diabetes.csv	5^{th}	0.4840	0.0432	0.8433
fanny	euclidean	4	diabetes.csv	6^{th}	0.4899	0.0431	0.8345
fanny	euclidean	4	diabetes.csv	7^{th}	0.5249	0.0104	0.8333
fanny	euclidean	4	diabetes.csv	8^{th}	0.5250	0.0052	0.8292
fanny	euclidean	4	diabetes.csv	9^{th}	0.5284	0.0013	0.8223
fanny	euclidean	4	diabetes.csv	10^{th}	0.6325	0.0000	0.0000
fanny	euclidean	4	preeclampsia.csv	1^{st}	0.0619	0.5024	0.4867

Tabla A.7: Resultados obtenidos por la librería *Clustering* para el algoritmo *Fanny*.

Algorithm	Distance	Clusters	Data	Var	Time	Precision	Recall
fanny	euclidean	4	preeclampsia.csv	2^{nd}	0.0621	0.5008	0.4713
fanny	euclidean	4	preeclampsia.csv	3^{rd}	0.0626	0.4984	0.4657
fanny	euclidean	4	preeclampsia.csv	4^{th}	0.0626	0.4984	0.4644
fanny	euclidean	4	preeclampsia.csv	5^{th}	0.0631	0.4980	0.4630
fanny	euclidean	4	preeclampsia.csv	6^{th}	0.0634	0.3352	0.4622
fanny	euclidean	4	preeclampsia.csv	7^{th}	0.0634	0.3344	0.4618
fanny	euclidean	4	preeclampsia.csv	8^{th}	0.0636	0.3316	0.4614
fanny	euclidean	4	preeclampsia.csv	9^{th}	0.0637	0.3309	0.4613
fanny	euclidean	4	preeclampsia.csv	10^{th}	0.0639	0.0569	0.4613
fanny	euclidean	4	preeclampsia.csv	11^{th}	0.0693	0.0495	0.4611
fanny	euclidean	4	preeclampsia.csv	12^{th}	0.0696	0.0490	0.4606
fanny	euclidean	4	preeclampsia.csv	13^{th}	0.0709	0.0300	0.4572
fanny	euclidean	4	preeclampsia.csv	14^{th}	0.0769	0.0152	0.4538
fanny	euclidean	4	preeclampsia.csv	15^{th}	0.0770	0.0055	0.4519
fanny	euclidean	4	preeclampsia.csv	16^{th}	0.0771	0.0000	0.0000
fanny	euclidean	4	preeclampsia.csv	17^{th}	0.0774	0.0000	0.0000
fanny	euclidean	4	preeclampsia.csv	18^{th}	0.0774	0.0000	0.0000
fanny	euclidean	4	preeclampsia.csv	19^{th}	0.0779	0.0000	0.0000
fanny	euclidean	4	preeclampsia.csv	20^{th}	0.0782	0.0000	0.0000
fanny	euclidean	4	preeclampsia.csv	21^{st}	0.0796	0.0000	0.0000
fanny	euclidean	4	preeclampsia.csv	22^{nd}	0.0798	0.0000	0.0000
fanny	euclidean	4	preeclampsia.csv	23^{rd}	0.0799	0.0000	0.0000
fanny	euclidean	4	preeclampsia.csv	24^{th}	0.0818	0.0000	0.0000
fanny	euclidean	4	preeclampsia.csv	25^{th}	0.0905	0.0000	0.0000
fanny	manhattan	3	diabetes.csv	1^{st}	0.4303	0.8391	0.5545
fanny	manhattan	3	diabetes.csv	2^{nd}	0.4368	0.2386	0.5270
fanny	manhattan	3	diabetes.csv	3^{rd}	0.4407	0.1331	0.4403
fanny	manhattan	3	diabetes.csv	4^{th}	0.4606	0.1038	0.4364
fanny	manhattan	3	diabetes.csv	5^{th}	0.4703	0.0645	0.3818
fanny	manhattan	3	diabetes.csv	6^{th}	0.4980	0.0476	0.3670
fanny	manhattan	3	diabetes.csv	7^{th}	0.5150	0.0123	0.3519
fanny	manhattan	3	diabetes.csv	8^{th}	0.5491	0.0054	0.3424
fanny	manhattan	3	diabetes.csv	9^{th}	0.6131	0.0012	0.3199
fanny	manhattan	3	diabetes.csv	10^{th}	0.7010	0.0000	0.0000
fanny	manhattan	3	preeclampsia.csv	1^{st}	0.0525	0.5020	1.0000
fanny	manhattan	3	preeclampsia.csv	2^{nd}	0.0526	0.5000	1.0000
fanny	manhattan	3	preeclampsia.csv	3^{rd}	0.0530	0.4988	1.0000
fanny	manhattan	3	preeclampsia.csv	4^{th}	0.0535	0.4988	1.0000
fanny	manhattan	3	preeclampsia.csv	5^{th}	0.0537	0.4988	1.0000
fanny	manhattan	3	preeclampsia.csv	6^{th}	0.0546	0.3353	1.0000
fanny	manhattan	3	preeclampsia.csv	7^{th}	0.0560	0.3329	1.0000
fanny	manhattan	3	preeclampsia.csv	8^{th}	0.0580	0.3329	1.0000
fanny	manhattan	3	preeclampsia.csv	9^{th}	0.0588	0.3318	1.0000
fanny	manhattan	3	preeclampsia.csv	10^{th}	0.0604	0.0571	1.0000
fanny	manhattan	3	preeclampsia.csv	11^{th}	0.0614	0.0501	1.0000
fanny	manhattan	3	preeclampsia.csv	12^{th}	0.0621	0.0501	1.0000

Tabla A.8: Resultados obtenidos por la librería *Clustering* para el algoritmo *Fanny*.

Algorithm	Distance	Clusters	Data	Var	Time	Precision	Recall
fanny	manhattan	3	preeclampsia.csv	13^{th}	0.0636	0.0294	1.0000
fanny	manhattan	3	preeclampsia.csv	14^{th}	0.0639	0.0155	1.0000
fanny	manhattan	3	preeclampsia.csv	15^{th}	0.0641	0.0052	1.0000
fanny	manhattan	3	preeclampsia.csv	16^{th}	0.0647	0.0000	0.0000
fanny	manhattan	3	preeclampsia.csv	17^{th}	0.0647	0.0000	0.0000
fanny	manhattan	3	preeclampsia.csv	18^{th}	0.0650	0.0000	0.0000
fanny	manhattan	3	preeclampsia.csv	19^{th}	0.0656	0.0000	0.0000
fanny	manhattan	3	preeclampsia.csv	20^{th}	0.0665	0.0000	0.0000
fanny	manhattan	3	preeclampsia.csv	21^{st}	0.0683	0.0000	0.0000
fanny	manhattan	3	preeclampsia.csv	22^{nd}	0.0761	0.0000	0.0000
fanny	manhattan	3	preeclampsia.csv	23^{rd}	0.0815	0.0000	0.0000
fanny	manhattan	3	preeclampsia.csv	24^{th}	0.0827	0.0000	0.0000
fanny	manhattan	3	preeclampsia.csv	25^{th}	0.0929	0.0000	0.0000
fanny	manhattan	4	diabetes.csv	1^{st}	0.6165	0.6679	0.8830
fanny	manhattan	4	diabetes.csv	2^{nd}	0.6270	0.2403	0.8702
fanny	manhattan	4	diabetes.csv	3^{rd}	0.6460	0.1208	0.8334
fanny	manhattan	4	diabetes.csv	4^{th}	0.6551	0.0998	0.8266
fanny	manhattan	4	diabetes.csv	5^{th}	0.6800	0.0481	0.7554
fanny	manhattan	4	diabetes.csv	6^{th}	0.6823	0.0436	0.7366
fanny	manhattan	4	diabetes.csv	7^{th}	0.6823	0.0110	0.7321
fanny	manhattan	4	diabetes.csv	8^{th}	0.6948	0.0052	0.7262
fanny	manhattan	4	diabetes.csv	9^{th}	0.7544	0.0013	0.7125
fanny	manhattan	4	diabetes.csv	10^{th}	0.8537	0.0000	0.0000
fanny	manhattan	4	preeclampsia.csv	1^{st}	0.0622	0.5020	1.0000
fanny	manhattan	4	preeclampsia.csv	2^{nd}	0.0632	0.5000	1.0000
fanny	manhattan	4	preeclampsia.csv	3^{rd}	0.0633	0.4988	1.0000
fanny	manhattan	4	preeclampsia.csv	4^{th}	0.0636	0.4988	1.0000
fanny	manhattan	4	preeclampsia.csv	5^{th}	0.0640	0.4988	1.0000
fanny	manhattan	4	preeclampsia.csv	6^{th}	0.0640	0.3353	1.0000
fanny	manhattan	4	preeclampsia.csv	7^{th}	0.0647	0.3329	1.0000
fanny	manhattan	4	preeclampsia.csv	8^{th}	0.0648	0.3329	1.0000
fanny	manhattan	4	preeclampsia.csv	9^{th}	0.0660	0.3318	1.0000
fanny	manhattan	4	preeclampsia.csv	10^{th}	0.0666	0.0571	1.0000
fanny	manhattan	4	preeclampsia.csv	11^{th}	0.0743	0.0501	1.0000
fanny	manhattan	4	preeclampsia.csv	12^{th}	0.0759	0.0501	1.0000
fanny	manhattan	4	preeclampsia.csv	13^{th}	0.0767	0.0294	1.0000
fanny	manhattan	4	preeclampsia.csv	14^{th}	0.0792	0.0155	1.0000
fanny	manhattan	4	preeclampsia.csv	15^{th}	0.0854	0.0052	1.0000
fanny	manhattan	4	preeclampsia.csv	16^{th}	0.0869	0.0000	0.0000
fanny	manhattan	4	preeclampsia.csv	17^{th}	0.0897	0.0000	0.0000
fanny	manhattan	4	preeclampsia.csv	18^{th}	0.0925	0.0000	0.0000
fanny	manhattan	4	preeclampsia.csv	19^{th}	0.0931	0.0000	0.0000
fanny	manhattan	4	preeclampsia.csv	20^{th}	0.1038	0.0000	0.0000
fanny	manhattan	4	preeclampsia.csv	21^{st}	0.1055	0.0000	0.0000
fanny	manhattan	4	preeclampsia.csv	22^{nd}	0.1074	0.0000	0.0000
fanny	manhattan	4	preeclampsia.csv	23^{rd}	0.1112	0.0000	0.0000
fanny	manhattan	4	preeclampsia.csv	24^{th}	0.1116	0.0000	0.0000
fanny	manhattan	4	preeclampsia.csv	25^{th}	0.1917	0.0000	0.0000

Tabla A.9: Resultados obtenidos por la librería *Clustering* para el algoritmo *Fanny*.

Ahora vamos a obtener el atributo del conjunto de datos con mejor calidad en el agrupamiento. Los métodos utilizados son: *Clustering::best_ranked_external_-metrics()* y *Clustering::best_ranked_internal_metrics()*. Los resultados quedan reflejados en las tablas A.10 y A.11.

Algorithm	Distance	Clusters	Data	Var	Time	Precision	Recall	TimeAtt	PrecisionAtt	RecallAtt
pam	euclidean	3	diabetes.csv	1st	0.0702	0.8958	0.5716	10th	10th	10th
pam	euclidean	3	preeclampsia.csv	1st	0.0165	0.5544	0.3883	8th	20th	22nd
pam	euclidean	4	diabetes.csv	1st	0.0820	0.8854	0.4797	10th	10th	10th
pam	euclidean	4	preeclampsia.csv	1st	0.0188	0.5654	0.2874	12th	18th	22nd
pam	manhattan	3	diabetes.csv	1st	0.0875	0.8138	0.5719	10th	10th	5th
pam	manhattan	3	preeclampsia.csv	1st	0.0159	0.6102	0.4100	10th	23rd	23rd
pam	manhattan	4	diabetes.csv	1st	0.0561	0.8568	0.4865	10th	10th	5th
pam	manhattan	4	preeclampsia.csv	1st	0.0179	0.5824	0.2931	8th	23rd	23rd
fanny	euclidean	3	diabetes.csv	1st	0.2960	0.6677	0.8270	10th	10th	9th
fanny	euclidean	3	preeclampsia.csv	1st	0.0446	0.5020	0.7942	1st	18th	4th
fanny	euclidean	4	diabetes.csv	1st	0.4556	0.5820	0.8934	8th	10th	3rd
fanny	euclidean	4	preeclampsia.csv	1st	0.0619	0.5024	0.4867	22nd	18th	11th
fanny	manhattan	3	diabetes.csv	1st	0.4303	0.8391	0.5545	4th	10th	9th
fanny	manhattan	3	preeclampsia.csv	1st	0.0525	0.5020	1.0000	12th	18th	1st
fanny	manhattan	4	diabetes.csv	1st	0.6165	0.6679	0.8830	9th	10th	10th
fanny	manhattan	4	preeclampsia.csv	1st	0.0622	0.5020	1.0000	18th	18th	1st
clara	euclidean	3	diabetes.csv	1st	0.0050	0.7801	0.7183	10th	10th	9th
clara	euclidean	3	preeclampsia.csv	1st	0.0036	0.5645	0.3876	11th	18th	22nd
clara	euclidean	4	diabetes.csv	1st	0.0051	0.7783	0.4423	10th	10th	5th
clara	euclidean	4	preeclampsia.csv	1st	0.0043	0.5609	0.3108	11th	18th	11th
clara	manhattan	3	diabetes.csv	1st	0.0048	0.7771	0.6955	10th	10th	9th
clara	manhattan	3	preeclampsia.csv	1st	0.0038	0.5632	0.3902	9th	18th	18th
clara	manhattan	4	diabetes.csv	1st	0.0047	0.6663	0.6806	10th	10th	5th
clara	manhattan	4	preeclampsia.csv	1st	0.0044	0.5804	0.3084	3rd	18th	18th

Tabla A.10: Evaluación de las medidas externas ordenados por algoritmo, medida de distancia y número de grupos.

Algorithm	Distance	Clusters	Data	Var	Time	Dunn	Silhouette	TimeAtt	DunnAtt	SilhouetteAtt
pam	euclidean	3	diabetes.csv	1st	0.0288	0.0535	0.12	9th	1st	1st
pam	euclidean	3	preeclampsia.csv	1st	0.0067	0.3966	0.03	4th	1st	1st
pam	euclidean	4	diabetes.csv	1st	0.0303	0.0628	0.12	10th	1st	1st
pam	euclidean	4	preeclampsia.csv	1st	0.0083	0.3966	0.02	25th	1st	1st
pam	manhattan	3	diabetes.csv	1st	0.0272	0.0584	0.14	6th	1st	1st
pam	manhattan	3	preeclampsia.csv	1st	0.0071	0.2856	0.03	8th	1st	1st
pam	manhattan	4	diabetes.csv	1st	0.0294	0.0714	0.15	6th	1st	1st
pam	manhattan	4	preeclampsia.csv	1st	0.0089	0.2868	0.03	4th	1st	1st
fanny	euclidean	3	diabetes.csv	1st	0.0256	0.0753	0.18	7th	1st	1st
fanny	euclidean	3	preeclampsia.csv	1st	0.0082	0.3591	0.00	2nd	1st	1st
fanny	euclidean	4	diabetes.csv	1st	0.0301	0.0629	-0.03	3rd	1st	1st
fanny	euclidean	4	preeclampsia.csv	1st	0.0081	0.3110	-0.02	22nd	1st	1st
fanny	manhattan	3	diabetes.csv	1st	0.0397	0.0470	0.10	2nd	1st	1st
fanny	manhattan	3	preeclampsia.csv	1st	0.0050	0.0000	0.00	3rd	1st	1st
fanny	manhattan	4	diabetes.csv	1st	0.0307	0.0649	0.19	7th	1st	1st
fanny	manhattan	4	preeclampsia.csv	1st	0.0061	0.0000	0.00	25th	1st	1st
clara	euclidean	3	diabetes.csv	1st	0.0260	0.0697	0.16	10th	1st	1st
clara	euclidean	3	preeclampsia.csv	1st	0.0083	0.3610	0.02	25th	1st	1st
clara	euclidean	4	diabetes.csv	1st	0.0342	0.0697	0.11	8th	1st	1st
clara	euclidean	4	preeclampsia.csv	1st	0.0091	0.3883	0.02	4th	1st	1st
clara	manhattan	3	diabetes.csv	1st	0.0341	0.0469	0.19	10th	1st	1st
clara	manhattan	3	preeclampsia.csv	1st	0.0078	0.2749	0.03	24th	1st	1st
clara	manhattan	4	diabetes.csv	1st	0.0347	0.0559	0.10	7th	1st	1st
clara	manhattan	4	preeclampsia.csv	1st	0.0084	0.2749	0.03	8th	1st	1st

Tabla A.11: Evaluación de las medidas internas ordenados por algoritmo, medida de distancia y número de grupos.

Además, la librería dispone de métodos para evaluar el comportamiento de las medidas de distancia en los algoritmos con el objetivo de reducir y facilitar el estudio de los resultados. Los resultados de evaluar la influencia de las medidas de distancia se observan en las tablas A.12 y A.13 para medidas externas y en las tablas A.14 y A.15 para las internas.

Algorithm	Distance	Clusters	Time	Precision	Recall	TimeAtt	PrecisionAtt	RecallAtt
pam	euclidean	3	0.0702	0.8958	0.5716	10^{th}	10^{th}	10^{th}
pam	manhattan	4	0.0561	0.8568	0.4865	10^{th}	10^{th}	5^{th}
fanny	euclidean	3	0.2960	0.6677	0.8270	10^{th}	10^{th}	9^{th}
fanny	manhattan	3	0.4303	0.8391	0.5545	4^{th}	10^{th}	9^{th}
clara	euclidean	3	0.0050	0.7801	0.7183	10^{th}	10^{th}	9^{th}
clara	manhattan	3	0.0048	0.7771	0.6955	10^{th}	10^{th}	9^{th}

Tabla A.12: Agrupación de los resultados por algoritmo y medida de distancia utilizando el método Clustering::evaluate_best_validation_external_by_metrics().

Algorithm	Distance	Clusters	Time	Precision	Recall	TimeAtt	PrecisionAtt	RecallAtt
pam	euclidean	3	0.0702	0.8958	0.5716	10^{th}	10^{th}	10^{th}
fanny	manhattan	3	0.4303	0.8391	0.5545	4^{th}	10^{th}	9^{th}
clara	euclidean	3	0.0050	0.7801	0.7183	10^{th}	10^{th}	9^{th}

Tabla A.13: Resultados de las medidas externas por algoritmo aplicando el método Clustering::result_external_algorithm_by_metric().

Algorithm	Distance	Clusters	Time	Dunn	Silhouette	TimeAtt	DunnAtt	SilhouetteAtt
pam	euclidean	3	0.0288	0.0535	0.12	9^{th}	1^{st}	1^{st}
pam	manhattan	4	0.0294	0.0714	0.15	6^{th}	1^{st}	1^{st}
fanny	euclidean	3	0.0256	0.0753	0.18	7^{th}	1^{st}	1^{st}
fanny	manhattan	4	0.0307	0.0649	0.19	7^{th}	1^{st}	1^{st}
clara	euclidean	3	0.0260	0.0697	0.16	10^{th}	1^{st}	1^{st}
clara	manhattan	3	0.0341	0.0469	0.19	10^{th}	1^{st}	1^{st}

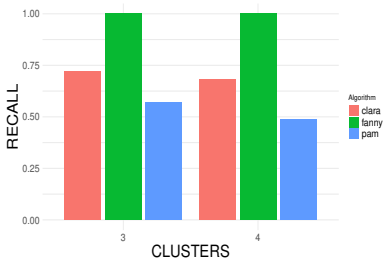
Tabla A.14: Agrupación de los resultados por algoritmo y medida de distancia utilizando el método Clustering::evaluate_best_validation_internal_by_metrics().

Algorithm	Distance	Clusters	Time	Dunn	Silhouette	TimeAtt	DunnAtt	SilhouetteAtt
pam	manhattan	4	0.0294	0.0714	0.15	6^{th}	1^{st}	1^{st}
fanny	manhattan	4	0.0307	0.0649	0.19	7^{th}	1^{st}	1^{st}
clara	manhattan	3	0.0341	0.0469	0.19	10^{th}	1^{st}	1^{st}

Tabla A.15: Resultados de las medidas internas por algoritmo aplicando el método Clustering::result_internal_algorithm_by_metric().

La librería incluye más métodos que permiten exportar los resultados a formato LaTeX y representar de forma gráfica los resultados, haciendo uso del método *Clustering::plot()*, tal y como queda plasmado en la figura A.1.

(a) Precision por número de grupos para cada algoritmo.



(b) Recall por número de grupos para cada algoritmo.

(c) Dunn por número de grupos para cada algoritmo.

(d) Silhouette por número de grupos para cada algoritmo.

Figura A.1: Representación gráfica de las medidas externa e interna por número de grupos y por algoritmo. *Fuente: elaboración propia.*

B
Apéndice

B.1. Tablas de resultados obtenidos por *CHCClust*

En esta sección se realiza un análisis comparativo entre varios algoritmos de agrupamiento ampliamente utilizados: clara, fanny, k-means, miniBatchKmeans y pam, y el algoritmo evolutivo *CHCClust*. Los resultados de cada celda se corresponden con el coeficiente de silueta. Se resalta en negrita aquellos valores de las celdas que tienen el mejor coeficiente de silueta por algoritmo y número de grupos. La información sobre los conjuntos de datos utilizados en este análisis se encuentran referenciados en la tabla 5.1.

		Grupos				
		3	4	5	6	7
Alcohol's effect on young people	clara	0.0150	0.0149	0.0098	0.0183	0.0155
	clara+*CHCClust*	**0.0218**	**0.0755**	**0.0842**	**0.0399**	**0.0625**
Basketball	clara	0.1730	0.1539	0.1097	0.1108	0.0398
	clara+*CHCClust*	**0.2404**	**0.2094**	**0.1537**	**0.1529**	**0.1546**
Bolts	clara	**0.2528**	**0.2146**	**0.2887**	0.2276	**0.1899**
	clara+*CHCClust*	0.2522	0.0656	0.1252	**0.2417**	0.0036
College	clara	0.2158	**0.1514**	0.1742	0.1524	**0.2689**
	clara+*CHCClust*	**0.2253**	0.1511	**0.1888**	**0.1630**	0.1562
ColorHistogram	clara	**0.0287**	**0.0197**	0.0050	0.0094	0.0081
	clara+*CHCClust*	0.0269	0.0153	**0.0196**	**0.0178**	**0.0102**
ColorMoments	clara	**0.1064**	**0.0900**	**0.0756**	**0.0659**	0.0263
	clara+*CHCClust*	0.0808	0.0811	0.0446	0.0508	**0.0694**
ColorTexture	clara	0.2134	0.1559	0.0635	0.0847	0.0620
	clara+*CHCClust*	**0.2525**	**0.1806**	**0.1501**	**0.1383**	**0.1346**
Country	clara	0.1822	0.1207	0.1105	**0.0619**	**0.1258**
	clara+*CHCClust*	**0.2036**	**0.1387**	**0.1580**	0.0462	0.0958
Diabetes	clara	**0.0676**	**0.0521**	0.0487	0.0452	0.0336
	clara+*CHCClust*	0.0586	0.0489	**0.0543**	**0.0551**	**0.0407**
Drug consumption	clara	0.1301	0.0651	0.0808	0.0745	**0.0511**
	clara+*CHCClust*	**0.1965**	**0.0812**	**0.1164**	**0.0941**	0.0501
Fetal health	clara	0.1251	0.0671	0.0768	0.1087	**0.1185**
	clara+*CHCClust*	**0.1290**	**0.0760**	**0.0775**	**0.1167**	0.1161
Haberman	clara	**0.0560**	**0.0824**	**0.0896**	**0.0151**	**0.0268**
	clara+*CHCClust*	0.0525	0.0513	0.0615	0.0023	-0.0056
Heart	clara	0.0046	**-0.0046**	**-0.0395**	**-0.0635**	-0.0453
	clara+*CHCClust*	**0.0218**	-0.0560	-0.0553	-0.0969	**0.0683**
Heart disease patients	clara	0.0352	0.0265	0.0142	0.0118	0.0047
	clara+*CHCClust*	**0.0447**	**0.0301**	**0.0191**	**0.0491**	**0.0291**
House16H	clara	**0.0799**	0.0458	**0.0451**	**0.0414**	0.0081
	clara+*CHCClust*	0.0680	**0.0501**	0.0382	0.0155	**0.0230**
India leads report	clara	0.2883	0.2831	**0.2377**	0.2650	**0.3501**
	clara+*CHCClust*	**0.3168**	**0.3155**	0.2368	**0.3004**	0.3004
Indicator districtwise health	clara	**0.1542**	0.1460	0.1533	0.1766	**0.1494**
	clara+*CHCClust*	0.1534	**0.1664**	**0.1579**	**0.1815**	0.1473
Iris	clara	**0.5579**	0.4523	**0.4078**	**0.4044**	**0.2270**
	clara+*CHCClust*	0.5552	**0.4609**	0.2939	0.2087	0.1874
LayoutHistogram	clara	0.0522	0.0467	**0.0495**	**0.0463**	**0.0431**
	clara+*CHCClust*	**0.0565**	**0.0595**	0.0456	0.0456	0.0362
Parkinson	clara	0.1315	0.0608	0.1024	0.0788	0.1240
	clara+*CHCClust*	**0.1802**	**0.1109**	**0.1694**	**0.1506**	**0.1915**
Pollution	clara	0.0252	**-0.0223**	-0.0982	-0.1034	-0.1147
	clara+*CHCClust*	**0.0331**	-0.0419	**-0.0419**	**0.0823**	**0.0303**
Price	clara	0.4302	0.3206	0.4004	**0.4004**	0.4104
	clara+*CHCClust*	**0.4335**	**0.3912**	**0.4042**	0.3994	**0.4064**
Quake	clara	**0.3020**	**0.2679**	**0.2150**	0.1580	0.1781
	clara+*CHCClust*	0.3010	0.2666	0.2114	**0.1581**	**0.1786**
Stulong	clara	**0.1427**	**0.1081**	**0.1099**	**0.1508**	**0.1324**
	clara+*CHCClust*	0.1416	0.0976	0.0938	**0.1388**	**0.1419**
Tae	clara	0.1615	**0.1583**	**0.1979**	**0.1452**	**0.1537**
	clara+*CHCClust*	**0.1632**	0.1517	0.1829	0.1292	0.0975
Transaction10K	clara	0.1563	0.1493	**0.1109**	0.0762	0.0854
	clara+*CHCClust*	**0.1943**	**0.1780**	0.1030	**0.1163**	**0.1190**
U.S.A. presidential results	clara	**0.0702**	0.0411	0.0184	0.0555	0.0398
	clara+*CHCClust*	0.0592	**0.1163**	**0.0863**	**0.1712**	**0.1451**
Vehicle	clara	0.1321	0.1456	0.1107	0.0890	0.0600
	clara+*CHCClust*	**0.1548**	**0.1660**	**0.1395**	**0.1046**	**0.0889**
Wine	clara	0.1411	0.0251	-0.0087	-0.0040	-0.0173
	clara+*CHCClust*	**0.2690**	**0.1645**	**0.1649**	**0.0779**	**0.0677**
Wholesale customers	clara	0.1569	0.1223	0.1123	0.0969	0.0589
	clara+*CHCClust*	**0.1840**	**0.1505**	**0.1926**	**0.1704**	**0.1151**

Tabla B.1: Resultados completos de comparar Clara y *CHCClust*.

		Grupos				
		3	4	5	6	7
Alcohol's effect on young people	fanny	0.0150	0.0148	0.0097	0.0183	0.0155
	fanny+*CHCClust*	**0.0247**	**0.0820**	**0.0821**	**0.0819**	**0.0820**
Basketball	fanny	**0.1890**	0.1261	0.1035	0.1261	0.0639
	fanny+*CHCClust*	0.1017	**0.2258**	**0.2223**	**0.2258**	**0.2329**
Bolts	fanny	0.2447	0.1733	**0.2372**	**0.2507**	0.0670
	fanny+*CHCClust*	**0.2799**	**0.2799**	0.0672	0.0147	**0.2210**
College	fanny	0.1698	**0.2017**	0.1698	0.1698	0.1698
	fanny+*CHCClust*	**0.1899**	0.1734	**0.1899**	**0.1899**	**0.1899**
ColorHistogram	fanny	0.0377	-0.0273	0.0043	**0.0100**	**0.0067**
	fanny+*CHCClust*	**0.0420**	**0.0134**	**0.0151**	-0.0050	-0.0128
ColorMoments	fanny	**0.1196**	0.0905	0.1024	**0.0885**	0.0904
	fanny+*CHCClust*	0.1164	**0.1622**	**0.1604**	0.0126	**0.1631**
ColorTexture	fanny	**0.2594**	0.1698	0.1310	0.1256	0.1198
	fanny+*CHCClust*	0.2885	**0.2075**	**0.1789**	**0.1945**	**0.1976**
Country	fanny	0.1822	0.1257	0.0876	0.0914	0.0648
	fanny+*CHCClust*	**0.2036**	**0.1388**	**0.1097**	**0.1238**	**0.1239**
Diabetes	fanny	**0.0640**	**0.0381**	0.0291	0.0185	0.0148
	fanny+*CHCClust*	0.0584	0.0341	**0.0298**	**0.0316**	**0.0356**
Drug consumption	fanny	0.1035	0.0096	0.0733	0.0395	**0.0790**
	fanny+*CHCClust*	**0.2040**	**0.0302**	**0.1065**	**0.1135**	**0.1026**
Fetal health	fanny	0.0730	0.1004	0.1227	0.1004	0.1483
	fanny+*CHCClust*	**0.1106**	**0.1514**	**0.1778**	**0.1514**	**0.1691**
Haberman	fanny	**0.0865**	**0.0290**	0.0038	**-0.0069**	**-0.0142**
	fanny+*CHCClust*	0.0856	0.0114	**0.0386**	-0.0100	-0.0500
Heart	fanny	0.0008	-0.0020	-0.0041	-0.0550	**-0.0113**
	fanny+*CHCClust*	**0.0669**	**0.0604**	**0.0567**	**0.0666**	-0.0737
Heart disease patients	fanny	0.0384	0.0212	0.0189	0.0189	0.0026
	fanny+*CHCClust*	**0.0427**	**0.0353**	**0.0442**	**0.1076**	**0.0266**
House16H	fanny	**0.0819**	**0.0545**	**0.0396**	**0.0266**	-0.0049
	fanny+*CHCClust*	0.0630	0.0303	0.0259	0.0152	**-0.0032**
India leads report	fanny	0.2773	**0.3996**	0.3074	0.3038	0.3205
	fanny+*CHCClust*	**0.2773**	0.3119	**0.3074**	**0.3038**	**0.3310**
Indicator districtwise health	fanny	0.2590	0.2590	0.2590	0.2590	0.2590
	fanny+*CHCClust*	**0.2703**	**0.2703**	**0.2703**	**0.2703**	**0.2703**
Iris	fanny	0.5286	0.4588	**0.3809**	0.1912	0.1715
	fanny+*CHCClust*	**0.5394**	**0.4227**	0.2510	**0.2307**	**0.249**3
LayoutHistogram	fanny	0.0518	0.0478	0.0445	**0.0429**	**0.0404**
	fanny+*CHCClust*	**0.0772**	**0.0507**	**0.0515**	0.0325	0.0327
Parkinson	fanny	0.0458	**0.0643**	0.0562	0.0938	0.0802
	fanny+*CHCClust*	**0.0489**	0.0632	**0.0621**	**0.1049**	**0.1702**
Pollution	fanny	0.0298	**-0.0223**	-0.0559	-0.0683	-0.0832
	fanny+ CHC	**0.0852**	-0.0419	**-0.0510**	**-0.0246**	**0.0352**
Price	fanny	0.3689	0.3531	0.2806	0.3398	0.3218
	fanny+*CHCClust*	**0.3697**	**0.3550**	**0.3387**	**0.3975**	**0.4014**
Quake	fanny	**0.3053**	**0.2658**	**0.2109**	0.1907	0.2039
	fanny+*CHCClust*	0.2347	0.1222	0.1971	**0.2126**	**0.2087**
Stulong	fanny	**0.0960**	**0.0530**	**0.0484**	**0.0103**	**0.0364**
	fanny+*CHCClust*	0.0143	-0.0542	-0.0332	-0.1340	-0.1193
Tae	fanny	**0.0911**	**0.0898**	**0.0907**	-0.1136	**0.0941**
	fanny+*CHCClust*	0.0204	-0.0541	0.0616	**-0.0286**	0.0527
Transaction10K	fanny	0.1844	0.1055	0.1352	**0.1683**	**0.1530**
	fanny+*CHCClust*	**0.1943**	**0.1576**	**0.1366**	0.0862	0.0674
U.S.A. presidential results	fanny	0.0435	0.0184	0.0089	0.0003	-0.0001
	fanny+*CHCClust*	**0.0664**	**0.0494**	**0.0167**	**0.0222**	**0.0097**
Vehicle	fanny	0.1341	0.1344	0.0510	0.0210	0.0435
	fanny+*CHCClust*	**0.1394**	**0.1594**	**0.1388**	**0.1278**	**0.1827**
Wine	fanny	0.1613	0.1330	**0.1179**	**0.0979**	**0.0967**
	fanny+*CHCClust*	**0.2846**	**0.1848**	0.0994	0.0841	0.0743
Wholesale customers	fanny	0.0374	0.0374	0.0374	0.0374	0.0374
	fanny+*CHCClust*	**0.0375**	**0.0375**	**0.0375**	**0.0375**	**0.0375**

Tabla B.2: Resultados completos de comparar Fanny y *CHCClust*.

		Grupos				
		3	4	5	6	7
Alcohol's effect on young people	kmeans	0.0183	0.0151	0.0170	0.0091	0.0099
	kmeans+*CHCClust*	**0.0257**	**0.0281**	**0.0598**	**0.0444**	**0.0301**
Basketball	k-means	0.1896	0.1589	0.1258	0.0073	0.0054
	k-means+*CHCClust*	**0.2209**	**0.1985**	**0.2128**	**0.1854**	**0.1469**
Bolts	k-means	0.2686	0.1749	**0.2749**	**0.2436**	**0.2076**
	k-means+*CHCClust*	**0.3673**	**0.2029**	0.0000	0.1156	0.0036
College	kmeans	0.2654	0.2005	0.1891	0.1652	0.1418
	kmeans+*CHCClust*	**0.2832**	**0.2204**	**0.2143**	**0.1837**	**0.1579**
ColorHistogram	k-means	0.0228	0.0212	**0.0133**	**0.0109**	**0.0083**
	k-means+*CHCClust*	**0.0367**	**0.0235**	0.0072	-0.0139	0.0009
ColorMoments	k-means	0.1237	0.0906	0.0866	0.0597	**0.0700**
	k-means+*CHCClust*	**0.1284**	**0.0958**	**0.1010**	**0.0758**	0.0685
ColorTexture	k-means	0.2278	0.1596	**0.1271**	0.0908	0.0992
	k-means+*CHCClust*	**0.2362**	**0.1783**	0.1244	**0.1291**	**0.1325**
Country	kmeans	0.1613	0.1394	0.1110	0.0845	0.0691
	kmeans+*CHCClust*	**0.1650**	**0.1650**	**0.1393**	**0.1049**	**0.0962**
Diabetes	kmeans	**0.0633**	**0.0636**	0.0409	0.0397	0.0313
	kmeans+*CHCClust*	0.0570	0.0626	**0.0496**	**0.0447**	**0.0378**
Drug consumption	kmeans	0.0966	0.0588	0.0578	0.0538	**0.0473**
	kmeans+*CHCClust*	**0.1073**	**0.0609**	**0.0588**	**0.0539**	0.0473
Fetal health	kmeans	0.1138	0.1202	0.1087	0.1272	0.1048
	kmeans+*CHCClust*	**0.1170**	**0.1317**	**0.1208**	**0.1369**	**0.1168**
Haberman	k-means	**0.0870**	0.0827	**0.0523**	0.0300	**-0.0070**
	k-means+*CHCClust*	0.0849	**0.0828**	0.0422	**0.0038**	-0.0111
Heart	k-means	**0.0108**	-0.0019	-0.0220	-0.0536	-0.0547
	k-means+*CHCClust*	-0.0009	**0.0417**	**0.0307**	**0.0156**	**0.0174**
Heart disease patients	kmeans	0.0426	0.0373	0.0194	0.0150	0.0090
	kmeans+*CHCClust*	**0.0498**	**0.0711**	**0.0331**	**0.0422**	**0.0458**
House16H	k-means	0.0343	0.0365	0.0380	0.0235	0.0215
	k-means+*CHCClust*	**0.0748**	**0.0795**	**0.0706**	**0.0694**	**0.0378**
India leads report	kmeans	0.3017	0.1143	0.1242	0.1967	0.2163
	kmeans+*CHCClust*	**0.3017**	**0.3249**	**0.1331**	**0.2679**	**0.2784**
Indicator districtwise health	kmeans	0.2253	0.1811	0.1555	0.1542	0.1348
	kmeans+*CHCClust*	**0.2271**	**0.1904**	**0.1589**	**0.1629**	**0.205**
Iris	k-means	**0.5364**	0.3326	0.2904	**0.4192**	0.1949
	k-means+*CHCClust*	0.5235	**0.4499**	**0.4077**	0.2941	**0.2230**
LayoutHistogram	k-means	**0.0595**	0.0426	**0.0513**	**0.0379**	**0.0418**
	k-means+*CHCClust*	0.0579	**0.0577**	0.0489	0.0037	0.0292
Parkinson	kmeans	0.1441	0.0615	0.0578	0.1058	0.072
	kmeans+*CHCClust*	**0.1960**	**0.1013**	**0.0843**	**0.1265**	**0.1229**
Pollution	k-means	**-0.0187**	**-0.0223**	-0.0524	-0.0581	-0.1364
	k-means+*CHCClust*	-0.0655	-0.0396	**0.1565**	**-0.0197**	**0.0470**
Price	k-means	0.3792	**0.4124**	0.3822	0.4024	0.3986
	k-means+*CHCClust*	**0.4688**	0.0417	**0.3854**	**0.4064**	**0.4322**
Quake	K-means	**0.3059**	**0.2693**	0.2181	**0.2132**	0.1864
	k-means+*CHCClust*	**0.3059**	0.1880	**0.2446**	0.1622	**0.2138**
Stulong	k-means	**0.2147**	**0.1546**	0.1344	0.1339	**0.1304**
	k-means+*CHCClust*	0.1459	0.1361	**0.1796**	**0.1570**	0.1251
Tae	k-means	0.1334	**0.1775**	**0.1676**	**0.1621**	0.1560
	k-means+*CHCClust*	**0.1836**	0.1003	0.0818	0.0512	**0.1570**
Transaction10K	K-means	0.1587	0.1322	0.1146	0.1032	0.0765
	k-means+*CHCClust*	**0.1943**	**0.1780**	**0.1339**	**0.1246**	**0.1109**
U.S.A. presidential results	kmeans	0.0512	0.0458	0.0210	0.0102	0.0098
	kmeans+*CHCClust*	**0.0974**	**0.0763**	**0.0549**	**0.0274**	**0.0201**
Vehicle	kmeans	0.1531	0.0646	**0.2445**	0.0618	**0.2126**
	kmeans+*CHCClust*	**0.1775**	**0.0719**	0.0656	**0.0809**	0.1148
Wine	k-means	0.1311	0.0107	-0.0080	-0.0057	-0.0268
	k-means+*CHCClust*	**0.2818**	**0.1836**	**0.1051**	**0.0863**	**0.0656**
Wholesale customers	kmeans	0.1611	0.1184	0.1185	0.1153	0.1172
	kmeans+*CHCClust*	**0.1770**	**0.1287**	**0.1944**	**0.1509**	**0.1482**

Tabla B.3: Resultados completos de comparar K-means y *CHCClust*.

		Grupos				
		3	4	5	6	7
Alcohol's effect on young people	pam	0.0142	0.0154	0.0103	0.0116	0.0119
	pam+*CHCClust*	**0.0185**	**0.0267**	**0.0281**	**0.0282**	**0.0412**
Basketball	pam	0.1653	0.1356	0.1085	0.1282	0.0598
	pam+*CHCClust*	**0.2212**	**0.1894**	**0.1411**	**0.1754**	**0.1729**
Bolts	pam	**0.2528**	**0.2146**	**0.2616**	0.2002	**0.1899**
	pam+*CHCClust*	0.2522	0.0656	0.2210	**0.2144**	0.0147
College	pam	0.2239	0.1547	0.1711	0.1495	0.1425
	pam+*CHCClust*	**0.2369**	**0.1619**	**0.1879**	**0.1630**	**0.1503**
ColorHistogram	pam	0.0355	**0.0285**	0.0126	**0.0102**	**0.0061**
	pam+*CHCClust*	**0.0375**	0.0054	**0.0196**	0.0053	0.0021
ColorMoments	pam	0.1022	**0.0873**	0.0824	0.0832	**0.0807**
	pam+*CHCClust*	**0.1127**	0.0771	**0.0737**	**0.0752**	0.0709
ColorTexture	pam	0.1973	0.1402	0.1135	0.0995	0.0809
	pam+*CHCClust*	**0.2767**	**0.1897**	**0.1587**	**0.1955**	**0.1411**
Country	pam	0.1646	0.1325	0.1153	0.1046	0.0853
	pam+*CHCClust*	**0.1943**	**0.1464**	**0.1356**	**0.1133**	**0.1626**
Diabetes	pam	**0.0662**	**0.0640**	0.0374	0.0381	0.0351
	pam+*CHCClust*	0.0614	0.0605	**0.0381**	**0.0434**	**0.0370**
Drug consumption	pam	0.1132	0.0664	0.0555	0.0444	0.0426
	pam+*CHCClust*	**0.1296**	**0.0822**	**0.0585**	**0.0505**	**0.0505**
Fetal health	pam	0.1266	0.1187	0.1301	0.1135	0.1011
	pam+*CHCClust*	**0.1307**	**0.1269**	**0.1338**	**0.1157**	**0.1053**
Haberman	pam	**0.0769**	**0.0828**	**0.0589**	0.0420	**0.0121**
	pam+*CHCClust*	0.0747	0.0772	-0.0551	**0.0507**	-0.0130
Heart	pam	0.0030	-0.0143	-0.0339	-0.0514	-0.0559
	pam+*CHCClust*	**0.0678**	**0.0600**	**0.0293**	**0.0293**	**0.0401**
Heart disease patients	pam	0.0343	0.0263	0.0176	0.0114	0.0086
	pam+*CHCClust*	**0.0547**	**0.0649**	**0.0511**	**0.0259**	**0.0194**
House16H	pam	0.0778	0.0478	**0.0388**	**0.0364**	0.0117
	pam+*CHCClust*	**0.0804**	**0.0535**	0.0338	0.0305	**0.0308**
India leads report	pam	0.2974	0.2831	**0.2377**	0.2310	0.2287
	pam+*CHCClust*	**0.3517**	**0.3195**	0.2368	**0.2772**	**0.2772**
Indicator districtwise health	pam	0.1824	0.1173	0.1283	0.1251	0.1106
	pam+*CHCClust*	**0.1909**	**0.1226**	**0.1493**	**0.1370**	**0.1251**
Iris	pam	**0.5579**	**0.4697**	0.4320	0.4096	0.2284
	pam+*CHCClust*	0.5552	0.4567	**0.4484**	0.4562	**0.2826**
LayoutHistogram	pam	0.0464	**0.0461**	0.0508	**0.0451**	**0.0399**
	pam+*CHCClust*	**0.0468**	0.0431	**0.0522**	0.0349	0.0357
Parkinson	pam	0.1279	0.0603	0.0986	0.1511	0.1286
	pam+*CHCClust*	**0.1793**	**0.1106**	**0.1666**	**0.2121**	**0.1838**
Pollution	pam	0.0298	-0.0223	-0.0534	-0.1034	-0.1141
	pam+*CHCClust*	**0.0736**	**0.0013**	**0.0823**	**0.0823**	**-0.0422**
Price	pam	**0.4688**	0.4108	0.4241	0.4271	**0.4212**
	pam+*CHCClust*	**0.4688**	**0.4161**	**0.4274**	**0.4345**	0.4204
Quake	pam	**0.3020**	**0.2680**	**0.2130**	**0.1912**	**0.1854**
	pam+*CHCClust*	0.3010	**0.2680**	0.2111	0.1890	0.1832
Stulong	pam	**0.1412**	**0.1282**	**0.1100**	**0.1440**	**0.1324**
	pam+*CHCClust*	0.1397	0.1119	0.0950	0.1340	0.1125
Tae	pam	**0.1630**	**0.1421**	**0.1948**	**0.1668**	**0.1664**
	pam+*CHCClust*	0.1439	0.0874	0.1827	0.1534	0.1037
Transaction10K	pam	0.1589	0.1300	0.1154	0.1054	0.0947
	pam+*CHCClust*	**0.1943**	**0.1780**	**0.1294**	**0.1218**	**0.1057**
U.S.A. presidential results	pam	0.0713	0.0534	0.0672	0.0497	0.0093
	pam+*CHCClust*	**0.1254**	**0.1287**	**0.1299**	**0.0962**	**0.0148**
Vehicle	pam	0.1268	**0.1019**	0.1056	0.1000	0.2208
	pam+*CHCClust*	**0.1268**	0.0944	**0.1121**	**0.1100**	**0.2526**
Wine	pam	0.1550	0.0186	-0.0056	-0.0040	-0.0169
	pam+*CHCClust*	**0.2768**	**0.1751**	**0.1615**	**0.0779**	**0.0759**
Wholesale customers	pam	0.1632	0.1290	0.0968	**0.0804**	0.0637
	pam+*CHCClust*	**0.1899**	**0.1540**	**0.1015**	0.0717	**0.0726**

Tabla B.4: Tabla de resultados con la comparación entre Pam y *CHCClust*.

		Grupos				
		3	4	5	6	7
Alcohol's effect on young people	miniBatchKmeans	0.0147	0.0137	**0.0149**	0.0154	**0.1623**
	miniBatchKmeans+*CHCClust*	**0.0388**	**0.0291**	0.0123	**0.0416**	0.1197
Basketball	miniBatchKmeans	0.1891	0.0473	0.0435	-0.0617	0.0273
	miniBatchKmeans+*CHCClust*	**0.2284**	**0.2021**	**0.1985**	**0.1971**	**0.1964**
Bolts	miniBatchKmeans	0.2686	**0.1773**	**0.2887**	**0.2114**	**0.2076**
	miniBatchKmeans+*CHCClust*	**0.3673**	0.0871	0.1252	0.1095	0.0535
College	miniBatchKmeans	0.3316	0.1434	**0.5453**	**0.3329**	**0.2675**
	miniBatchKmeans+*CHCClust*	**0.3609**	**0.2680**	0.2680	0.2684	0.2115
ColorHistogram	miniBatchKmeans	0.0091	0.0035	-0.0003	-0.0025	**-0.0048**
	miniBatchKmeans+*CHCClust*	**0.0105**	**0.0158**	**0.0034**	**0.0024**	-0.0142
ColorMoments	miniBatchKmeans	0.0138	0.0041	0.0012	-0.0032	-0.0019
	miniBatchKmeans+*CHCClust*	**0.0866**	**0.0810**	**0.0325**	**0.0329**	**0.0194**
ColorTexture	miniBatchKmeans	-0.0124	-0.0146	-0.0255	-0.0234	-0.0278
	miniBatchKmeans+*CHCClust*	**0.2628**	**0.2676**	**0.0588**	**0.0696**	**0.0851**
Country	miniBatchKmeans	0.1638	0.1619	0.1267	**0.2838**	0.2221
	miniBatchKmeans+*CHCClust*	**0.1759**	**0.1778**	**0.2469**	0.2826	**0.2612**
Diabetes	miniBatchKmeans	0.0643	0.0765	0.0631	0.0361	0.0401
	miniBatchKmeans+*CHCClust*	**0.0662**	**0.0869**	**0.0660**	**0.0399**	**0.0584**
Drug consumption	miniBatchKmeans	0.0920	0.0655	0.0753	0.0729	0.0844
	miniBatchKmeans+*CHCClust*	**0.1835**	**0.0859**	**0.0994**	**0.1100**	**0.0994**
Fetal health	miniBatchKmeans	0.1302	0.1283	0.1259	0.0912	**0.0962**
	miniBatchKmeans+*CHCClust*	**0.1570**	**0.1417**	**0.1322**	**0.1050**	0.0930
Haberman	miniBatchKmeans	**0.1305**	**0.0833**	**0.0828**	**0.0422**	**-0.0090**
	miniBatchKmeans+*CHCClust*	0.1109	0.0521	0.0071	0.0248	-0.0192
Heart	miniBatchKmeans	**-0.0067**	-0.0347	-0.0499	-0.0581	-0.0562
	miniBatchKmeans+*CHCClust*	-0.0121	**0.0271**	**0.0931**	**0.0770**	**-0.0247**
Heart disease patients	miniBatchKmeans	0.0408	0.0464	0.0374	0.0231	0.0148
	miniBatchKmeans+*CHCClust*	**0.0606**	**0.0986**	**0.1415**	**0.0923**	**0.0493**
House16H	miniBatchKmeans	0.0514	0.0418	-0.0010	-0.0022	-0.0056
	miniBatchKmeans+*CHCClust*	**0.0790**	**0.0657**	**0.0286**	**0.0177**	**0.0145**
India leads report	miniBatchKmeans	0.2717	0.2522	0.2533	0.2815	**0.4211**
	miniBatchKmeans+*CHCClust*	**0.3168**	**0.2648**	**0.2843**	**0.3434**	0.3258
Indicator districtwise health	miniBatchKmeans	0.2452	0.1484	0.1417	0.1410	0.1235
	miniBatchKmeans+*CHCClust*	**0.2494**	**0.1841**	**0.1654**	**0.1472**	**0.1782**
Iris	miniBatchKmeans	0.5003	**0.4530**	0.3653	**0.3410**	**0.2475**
	miniBatchKmeans+*CHCClust*	**0.5250**	0.4035	**0.3942**	0.2262	0.2295
LayoutHistogram	miniBatchKmeans	0.0044	-0.0003	-0.0037	-0.0042	-0.0052
	miniBatchKmeans+*CHCClust*	**0.0263**	**0.0413**	**-0.0024**	**0.0233**	**0.0216**
Parkinson	miniBatchKmeans	0.1446	0.1514	0.1362	0.1372	0.1191
	miniBatchKmeans+*CHCClust*	**0.2307**	**0.2289**	**0.1853**	**0.2030**	**0.2902**
Pollution	miniBatchKmeans	**-0.0087**	-0.1039	-0.1289	-0.1285	-0.1409
	miniBatchKmeans+*CHCClust*	-0.0923	**-0.0923**	**0.1355**	**0.1186**	**0.0823**
Price	miniBatchKmeans	0.4625	0.4027	0.4005	0.3949	0.3306
	miniBatchKmeans+*CHCClust*	**0.4640**	**0.4117**	**0.4163**	**0.4112**	**0.4416**
Quake	miniBatchKmeans	**0.2988**	0.1676	0.1499	**0.1349**	0.1308
	miniBatchKmeans+*CHCClust*	0.2965	**0.1679**	**0.1512**	0.1192	0.1144
Stulong	miniBatchKmeans	**0.1309**	0.1486	**0.1379**	**0.1375**	**0.1078**
	miniBatchKmeans+*CHCClust*	0.1270	**0.1502**	0.1353	0.1311	**0.1078**
Tae	miniBatchKmeans	**0.1473**	0.1338	0.1863	**0.1282**	**0.0795**
	miniBatchKmeans+*CHCClust*	0.1379	**0.1451**	**0.1650**	**0.1264**	0.0560
Transaction10K	miniBatchKmeans	0.1209	0.0891	0.1184	0.0756	**0.1194**
	miniBatchKmeans+*CHCClust*	**0.1779**	**0.1780**	**0.1518**	**0.0980**	0.0621
U.S.A. presidential results	miniBatchKmeans	0.0512	0.0347	0.0212	0.0198	0.0094
	miniBatchKmeans+*CHCClust*	**0.0835**	**0.1261**	**0.1354**	**0.1672**	**0.1532**
Vehicle	miniBatchKmeans	0.1355	**0.1478**	**0.3012**	**0.2556**	**0.2800**
	miniBatchKmeans+*CHCClust*	**0.1503**	0.1336	0.1702	0.1233	0.2016
Wine	miniBatchKmeans	0.1137	-0.0015	-0.0211	-0.0396	-0.0260
	miniBatchKmeans+*CHCClust*	**0.2671**	**0.1428**	**0.0213**	**0.1305**	**0.1602**
Wholesale customers	miniBatchKmeans	0.1717	0.1360	0.1957	0.1519	0.1082
	miniBatchKmeans+*CHCClust*	**0.3018**	**0.2338**	**0.2108**	**0.1752**	**0.1420**

Tabla B.5: Tabla de resultados con la comparativa entre los algoritmos MinibatchKmenas y *CHCClust*.

En base a los resultados de las tablas B.1, B.2, B.3, B.4 y B.5 se puede concluir que:

- *CHCClust* tiende a mejorar los valores del coeficiente de silueta en la mayoría de los conjuntos de datos y configuraciones de número de grupos. Esto indica que la aplicación de *CHCClust* sobre los algoritmos de agrupamiento generalmente produce grupos compactos y bien separados. El porcentaje de mejora está por encima del 70 %, alcanzando la tasa más alta para los algoritmos k-means y miniBatchKmeans.
- *CHCClust* tiende reducir la variaciones en la calidad del agrupamiento ocasionado por la búsqueda de centroides, proporcionando resultados más predecibles y fiables.
- Otro de los aspectos de *CHCClust* es la robustez debido a su capacidad de adaptarse y mejorar el agrupamiento para diferentes contextos y tipos de datos.
- En las configuraciones donde el número de grupos es bajo, *CHCClust* consigue un agrupamiento más óptimo.

En conclusión, *CHCClust* ha demostrado su capacidad para optimizar problemas complejos y mejorar los agrupamientos a través de la optimización de los grupos mediante el concepto de hiper-rectángulo y el algoritmo evolutivo CHC. Esta combinación de técnicas permite a *CHCClust* abordar eficazmente la diversidad y la complejidad presente en diferentes tipos de datos y entornos.

C
Apéndice

C.1. Tabla de resultados obtenidos por *MultiCHCClust*

En esta sección se presentan los resultados obtenidos de comparar el rendimiento del algoritmo *MultiCHCClust* con algoritmos de agrupamiento tradicionales. Utilizamos los conjuntos de datos definidos en la tabla 5.1. Cada fila de la tabla muestra el coeficiente de silueta por algoritmo y grupos, destacando en negrita aquellos valores que tienen mejor agrupamiento. El número de grupos utilizados es $k = 3$.

Los resultados demuestran que el algoritmo *MultiCHCClust* obtiene los valores más altos en términos de calidad en la mayoría de los conjuntos de datos evaluados. Es destacable su superioridad en 22 de los 30 conjuntos de datos analizados, reflejando su robustez y efectividad en la agrupación de diferentes tipos de datos.

Además, los conjuntos de datos utilizados son muy variados. Esto demuestra un comportamiento equilibrado de *MultiCHCClust* en comparación con otros algoritmos, donde en ciertas situaciones algunos algoritmos mejoran su comportamiento cuando trabaja con ciertos volúmenes de datos.

En resumen, he de destacar que *MultiCHCClust* es una opción robusta y competitiva en términos de precisión y adaptabilidad a diferentes escenarios de agrupamiento de datos.

	Algoritmos					
	Clara	Fanny	K-means	Pam	MiniBatchKmeans	*MultiCHCClust*
Alcohol's effect on young people	0.0150	0.0150	0.0183	0.0142	0.0147	**0.0387**
Basketball	0.1730	0.1890	0.1896	0.1653	0.1891	**0.2964**
Bolts	0.2528	0.2447	0.2686	0.2528	0.2686	**0.3673**
College	0.2158	0.1698	0.2654	0.2239	0.3316	**0.3494**
ColorHistogram	0.0287	**0.0377**	0.0228	0.0355	0.0091	0.0251
ColorMoments	0.1064	0.1196	**0.1237**	0.1022	0.0138	0.0956
ColorTexture	0.2134	0.2594	0.2278	0.1973	-0.0124	**0.2645**
Country	0.1822	0.1822	0.1613	0.1646	0.1638	**0.2036**
Diabetes	0.0676	0.0640	0.0633	0.0662	0.0643	**0.0859**
Drug consumption	0.1301	0.1035	0.0966	0.1132	0.0920	**0.2194**
Fetal health	0.1251	0.0730	0.1138	0.1266	0.1302	**0.1714**
Haberman	0.0560	0.0865	0.0870	0.0769	**0.1305**	0.0641
Heart	0.0046	0.0008	0.0108	0.0030	-0.0067	**0.0405**
Heart disease patients	0.0352	0.0384	0.0426	0.0343	0.0408	**0.0454**
House16H	0.0799	0.0819	0.0343	0.0778	0.0514	**0.0930**
India leads report	0.2883	0.2773	0.3017	0.2974	0.2717	**0.3716**
Indicator districtwise health	0.1542	0.2590	0.2253	0.1824	0.1484	**0.2602**
Iris	**0.5579**	0.5286	0.5364	0.5579	0.5003	0.5552
LayoutHistogram	0.0522	0.0518	**0.0595**	0.0464	0.0044	0.0532
Parkinson	0.1315	0.0458	0.1441	0.1279	0.1446	**0.1801**
Pollution	0.0252	0.0298	-0.0187	0.0298	-0.0087	**0.0852**
Price	0.4302	0.3689	0.3792	**0.4688**	0.4625	0.4500
Quake	0.3020	0.3053	**0.3059**	0.3020	0.2988	0.3010
Stulong	0.1427	0.0960	**0.2147**	0.1412	0.1309	0.1396
Tae	0.1615	0.0911	0.1334	0.1630	0.1473	**0.1871**
Transaction10K	0.1563	0.1844	0.1587	0.1589	0.1209	**0.1943**
U.S.A. presidential results	0.0702	0.0435	0.0512	0.0713	0.0347	**0.0834**
Vehicle	0.1321	0.1341	0.1531	0.1268	0.1355	**0.1540**
Wine	0.1411	0.1613	0.1311	0.1550	0.1137	**0.2940**
Wholesale customers	0.1569	0.0374	0.1611	0.1632	0.1717	**0.1917**
Mejor	1	1	4	1	1	22

Tabla C.1: Resultados del estudio experimental para el algoritmo *MultiCHCClust*.

Bibliografía

Abdullah Karaaslanlı y Selin Aviyente (2021). «Community detection in dynamic networks: Equivalence between stochastic blockmodels and evolutionary spectral clustering». En: IEEE Transactions on Signal and Information Processing over Networks 7, págs. 130-143.

Absalom E Ezugwu (2020). «Nature-inspired metaheuristic techniques for automatic clustering: a survey and performance study». En: SN Applied Sciences 2, págs. 1-57.

Absalom E Ezugwu, Amit K Shukla, Moyinoluwa B Agbaje, Olaide N Oyelade, Adán José-García y Jeffery O Agushaka (2021). «Automatic clustering algorithms: a systematic review and bibliometric analysis of relevant literature». En: Neural Computing and Applications 33, págs. 6247-6306.

Adam Slowik (2010). «Application of an adaptive differential evolution algorithm with multiple trial vectors to artificial neural network training». En: IEEE Transactions on Industrial Electronics 58(8), págs. 3160-3167.

Adán José-García y Wilfrido Gómez-Flores (2016). «Automatic clustering using nature-inspired metaheuristics: A survey». En: Applied Soft Computing 41, págs. 192-213.

AFL Nemec y RO Brinkhurst (1988). «The Fowlkes–Mallows statistic and the comparison of two independently determined dendrograms». En: Canadian Journal of Fisheries and Aquatic Sciences 45(6), págs. 971-975.

Agoston E Eiben y James E Smith (2015). *Introduction to evolutionary computing*. Springer.

Aidin Delgoshaei y A Ali (2019). «Evolution of clustering techniques in designing cellular manufacturing systems: A state-of-art review». En: International Journal of Industrial Engineering Computations 10(2), págs. 177-198.

Bibliografía

Alexandros Sfyridis y Paolo Agnolucci (2020). «Annual average daily traffic estimation in England and Wales: An application of clustering and regression modelling». En: Journal of Transport Geography 83, pág. 102658.

Amir Ahmad y Shehroz S Khan (2019). «Survey of state-of-the-art mixed data clustering algorithms». En: Ieee Access 7, págs. 31883-31902.

Amit Saxena, Mukesh Prasad, Akshansh Gupta, Neha Bharill, Om Prakash Patel, Aruna Tiwari, Meng Joo Er, Weiping Ding y Chin-Teng Lin (2017). «A review of clustering techniques and developments». En: Neurocomputing 267, págs. 664-681.

Anbupalam Thalamuthu, Indranil Mukhopadhyay, Xiaojing Zheng y George Tseng (2006). «Evaluation and comparison of gene clustering methods in microarray analysis». En: Bioinformatics 22(19), págs. 2405-2412.

Anil K Jain, M Narasimha Murty y Patrick J Flynn (1999). «Data clustering: a review». En: ACM computing surveys (CSUR) 31(3), págs. 264-323.

Anja Struyf, Mia Hubert y Peter Rousseeuw (1997). «Clustering in an object-oriented environment». En: Journal of Statistical Software 1, págs. 1-30.

Anton V Ushakov e Igor Vasilyev (2021). «Near-optimal large-scale k-medoids clustering». En: Information Sciences 545, págs. 344-362.

Anwar Said, Rabeeh Ayaz Abbasi, Onaiza Maqbool, Ali Daud y Naif Radi Aljohani (2018). «CC-GA: A clustering coefficient based genetic algorithm for detecting communities in social networks». En: Applied Soft Computing 63, págs. 59-70.

Archana Singh, Avantika Yadav y Ajay Rana (2013). «K-means with Three different Distance Metrics». En: International Journal of Computer Applications 67(10).

Arthur L. Samuel (1967). «Some Studies in Machine Learning Using the Game of Checkers». En: IBM J. Res. Dev. 44, págs. 206-227.

Artur Starczewski y Adam Krzyżak (2015). «Performance evaluation of the silhouette index». En: *Artificial Intelligence and Soft Computing: 14th International Conference, ICAISC 2015, Zakopane, Poland, June 14-18, 2015, Proceedings, Part II 14*, págs. 49-58.

Arvinder Kaur, Saibal Pal y Amrit Singh (2017). «Hybridization of K-Means and Firefly Algorithm for intrusion detection system». En: International Journal of System Assurance Engineering and Management 9, págs. 1-10.

Ashish Kumar Patnaik, Prasanta Kumar Bhuyan y KV Krishna Rao (2016). «Divisive Analysis (DIANA) of hierarchical clustering and GPS data for level of service criteria of urban streets». En: Alexandria Engineering Journal 55(1), págs. 407-418.

Bing Zhou, Bei Lu y Salman Saeidlou (2024). «A hybrid clustering method based on the several diverse basic clustering and meta-clustering aggregation technique». En: Cybernetics and Systems 55(1), págs. 203-229.

Biswarup Ray, Soulib Ghosh, Shameem Ahmed, Ram Sarkar y Mita Nasipuri (2022). «Outlier detection using an ensemble of clustering algorithms». En: Multimedia Tools and Applications, págs. 1-29.

Blaise Hanczar y Mohamed Nadif (2013). «Precision-recall space to correct external indices for biclustering». En: *International Conference on Machine Learning*, págs. 136-144.

Brian Everitt (1980). «Cluster analysis.» En: Quality & Quantity 14(1).

Bryar A Hassan, Tarik A Rashid y Hozan K Hamarashid (2021). «A novel cluster detection of COVID-19 patients and medical disease conditions using improved evolutionary clustering algorithm star». En: Computers in biology and medicine 138, pág. 104866.

ChemsEddine Berbague, Nour El Islem Karabadji y Hassina Seridi (2018). «An Evolutionary Scheme for Improving Recommender System Using Clustering». En: *Computational Intelligence and Its Applications*, págs. 290-301.

Chih-Chin Lai y Chuan-Yu Chang (2009). «A hierarchical evolutionary algorithm for automatic medical image segmentation». En: Expert Systems with Applications 36, págs. 248-259.

Chih-Hsun Chou, Su-Chen Hsieh y Chui-Jie Qiu (2017). «Hybrid Genetic Algorithm and Fuzzy Clustering for Bankruptcy Prediction». En: Applied Soft Computing 56(C), págs. 298-316.

Cor J Veenman, Marcel JT Reinders y Eric Backer (2003). «A cellular coevolutionary algorithm for image segmentation». En: IEEE Transactions on Image Processing 12(3), págs. 304-316.

D. E. Goldberg (1989). *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison-Wesley Longman Publishing Co., Inc.

Daniel Gribel y Thibaut Vidal (2019). «HG-means: A scalable hybrid genetic algorithm for minimum sum-of-squares clustering». En: Pattern Recognition 88, págs. 569-583.

Daniel J Feller, Marissa Burgermaster, Matthew E Levine, Arlene Smaldone, Patricia G Davidson, David J Albers y Lena Mamykina (2018). «A visual analytics approach for pattern-recognition in patient-generated data». En: Journal of the American Medical Informatics Association 25(10), págs. 1366-1374.

David Gabriel de Barros Franco y Maria Teresinha Arns Steiner (2018). «Clustering of solar energy facilities using a hybrid fuzzy c-means algorithm initialized by metaheuristics». En: Journal of Cleaner Production 191, págs. 445-457.

Dingsheng Deng (2020). «DBSCAN clustering algorithm based on density». En: *2020 7th international forum on electrical engineering and automation (IFEEA)*, págs. 949-953.

Dinkar Sitaram, Aditya Dalwani, Anish Narang, Madhura Das y Prafullata Auradkar (2015). «A measure of similarity of time series containing missing data using the mahalanobis distance». En: *2015 second international conference on advances in computing and communication engineering*, págs. 622-627.

Dongkuan Xu y Yingjie Tian (2015). «A comprehensive survey of clustering algorithms». En: Annals of data science 2, págs. 165-193.

Dongwei Guo, Jingjing Zhao y Jici Liu (2019). «Research and application of improved CHAMELEON algorithm based on condensed hierarchical clustering method». En: *Proceedings of the 2019 8th International Conference on Networks, Communication and Computing*, págs. 14-18.

Douglas M Hawkins (1980). *Identification of outliers*. (11).

Eden WM Ma y Tommy WS Chow (2004). «A new shifting grid clustering algorithm». En: Pattern recognition 37(3), págs. 503-514.

Emrah Hancer y Dervis Karaboga (2017). «A comprehensive survey of traditional, merge-split and evolutionary approaches proposed for determination of cluster number». En: Swarm and Evolutionary Computation 32, págs. 49-67.

Enrique H Ruspini, James C Bezdek y James M Keller (2019). «Fuzzy clustering: A historical perspective». En: IEEE Computational Intelligence Magazine 14(1), págs. 45-55.

Eréndira Rendón, Itzel Abundez, Alejandra Arizmendi y Elvia M Quiroz (2011). «Internal versus external cluster validation indexes». En: International Journal of computers and communications 5(1), págs. 27-34.

Fabrizio Cerreto, Bo Friis Nielsen, Otto Nielsen y Steven Harrod (abr. de 2018). «Application of Data Clustering to Railway Delay Pattern Recognition». En: Journal of Advanced Transportation 2018, págs. 1-18.

Faisal Ramzan y Muawaz Ayyaz (2023). «A comprehensive review on data stream mining techniques for data classification; and future trends». En: EPH-International Journal of Science And Engineering 9(3), págs. 1-29.

Frank Nielsen y Frank Nielsen (2016). «Hierarchical clustering». En: Introduction to HPC with MPI for Data Science, págs. 195-211.

Frank Wilcoxon (1992). «Individual comparisons by ranking methods». En: *Breakthroughs in statistics: Methodology and distribution*, págs. 196-202.

Gang Wang, Jinxing Hao, Jian Ma y Lihua Huang (2010). «A new approach to intrusion detection using Artificial Neural Networks and fuzzy clustering». En: Expert Systems with Applications 37, págs. 6225-6232.

Giulio Vialetto y Marco Noro (2020). «An innovative approach to design cogeneration systems based on big data analysis and use of clustering methods». En: Energy Conversion and Management 214, pág. 112901.

GN Lance y WT Williams (1966). «A generalized sorting strategy for computer classifications». En: Nature 212(5058), págs. 218-218.

GS Nithya y K Arun Prabha (2019). «A lion optimization based k-prototype clustering algorithm for mixed data.» En: system 6(02).

Gulezar Shamim y Mohd Rihan (2020). «Multi-domain feature extraction for improved clustering of smart meter data». En: Technology and Economics of Smart Grids and Sustainable Energy 5, págs. 1-8.

Guoxian Yu, Liangrui Ren, Jun Wang, Carlotta Domeniconi y Xiangliang Zhang (2024). «Multiple clusterings: Recent advances and perspectives». En: Computer Science Review 52, pág. 100621.

H Ramprasanth y A Devi (2019). «Outlier analysis of medical dataset using clustering algorithms». En: J. Anal. Comput 15, págs. 1-9.

H. Liu, J. Li y M. A. Chapman (2015). «Automated Road Extraction from Satellite Imagery Using Hybrid Genetic Algorithms and Cluster Analysis». En: Journal of Enviromental Informatics 1(2).

Hajar Khalili, Mohsen Rabbani y Ebrahim Akbari (2021). «Clustering ensemble selection based on the extended Jaccard measure». En: Turkish Journal of Electrical Engineering and Computer Sciences 29(4), págs. 2215-2231.

Hana Řezanková y B Everitt (2009). «Cluster analysis and categorical data». En: Statistika 89(3), págs. 216-232.

Hanan G Ayad y Mohamed S Kamel (2010). «On voting-based consensus of cluster ensembles». En: Pattern Recognition 43(5), págs. 1943-1953.

Hannah Johns, John Hearne, Julie Bernhardt y Leonid Churilov (2020). «Clustering clinical and health care processes using a novel measure of dissimilarity for variable-length sequences of ordinal states». En: Statistical methods in medical research 29(10), págs. 3059-3075.

Hong Li, Li Zhang y Yongchang Jiao (2016). «Discrete differential evolution algorithm for integer linear bilevel programming problems». En: Journal of Systems Engineering and Electronics 27(4), págs. 912-919.

Hyunsoo Kim y Haesun Park (2007). «Sparse non-negative matrix factorizations via alternating non-negativity-constrained least squares for microarray data analysis». En: Bioinformatics 23(12), págs. 1495-1502.

Bibliografía

Israel Edem Agbehadji, Richard C Millham, Abdultaofeek Abayomi, Jason J Jung, Simon James Fong y Samuel Ofori Frimpong (2021). «Clustering algorithm based on nature-inspired approach for energy optimization in heterogeneous wireless sensor network». En: Applied Soft Computing 104, pág. 107171.

James MacQueen et al. (1967). «Some methods for classification and analysis of multivariate observations». En: *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*. (1). 14, págs. 281-297.

James C Bezdek y Nikhil R Pal (1995). «Cluster validation with generalized Dunn's indices». En: *Proceedings 1995 second New Zealand international two-stream conference on artificial neural networks and expert systems*, págs. 190-193.

James C Bezdek, Robert Ehrlich y William Full (1984). «FCM: The fuzzy c-means clustering algorithm». En: Computers & geosciences 10(2-3), págs. 191-203.

Jasmine Irani, Nitin Pise y Madhura Phatak (2016). «Clustering techniques and the similarity measures used in clustering: A survey». En: International journal of computer applications 134(7), págs. 9-14.

Jianrui Chen, Uliji, Hua Wang y Zaizai Yan (2018). «Evolutionary heterogeneous clustering for rating prediction based on user collaborative filtering». En: Swarm and Evolutionary Computation 38, págs. 35-41.

Jingyou Zhang y Haiping Zhong (2022). «Analysis of Clustering Algorithms in Machine Learning for Healthcare Data». En: Journal of Commercial Biotechnology 27(5), págs. 82-91.

Jinyan Lu, Albert Gan, Kirolos Haleem y Wanyang Wu (2013). «Clustering-based roadway segment division for the identification of high-crash locations». En: Journal of Transportation Safety & Security 5(3), págs. 224-239.

John H Holland (1975). *Adaptation in natural and artificial systems*. University of Michigan Press.

John R Koza (1994). «Genetic programming as a means for programming computers by natural selection». En: Statistics and computing 4, págs. 87-112.

José E Chacón y Ana I Rastrojo (2023). «Minimum adjusted Rand index for two clusterings of a given size». En: Advances in Data Analysis and Classification 17(1), págs. 125-133.

José M Mazón Ruiz (2011). *Cálculo diferencial: teoría y problemas*. Universitat de València.

Junjie Wu, Hui Xiong y Jian Chen (2009). «Towards understanding hierarchical clustering: A data distribution perspective». En: Neurocomputing 72(10-12), págs. 2319-2330.

Kamran Khan, Saif Ur Rehman, Kamran Aziz, Simon Fong y Sababady Sarasvady (2014). «DBSCAN: Past, present and future». En: *The fifth international conference on the applications of digital information and web technologies (ICADIWT 2014)*, págs. 232-238.

Klaus Backhaus, Bernd Erichson, Sonja Gensler, Rolf Weiber y Thomas Weiber (2021). «Multivariate Analysis, An Application-Oriented Introduction». En: págs. 451-530.

Kristina P Sinaga y Miin-Shen Yang (2020). «Unsupervised K-means clustering algorithm». En: IEEE access 8, págs. 80716-80727.

L KaufmanandP y J Rousseeuw (1990). «Finding groups in data: an introduction to cluster analysis». En: JohnWiley & Sons.

L Jegatha Deborah, R Baskaran y A Kannan (2010). «A survey on internal validity measure for cluster validation». En: International Journal of Computer Science & Engineering Survey 1(2), págs. 85-102.

L.A. Pérez-Martos, A.M. García-Vico, P. González y C.J. Carmona (2022). «A Case of Study with the Clustering R Library to Measure the Quality of Cluster Algorithms». En: *International Conference on Hybrid Artificial Intelligence Systems*, págs. 88-97.

—— (2023a). «A Multiclustering Evolutionary Hyperrectangle-Based Algorithm». En: International Journal of Computational Intelligence Systems 16(1), pág. 165.

—— (2023b). «An Evolutionary Fuzzy System for Multiclustering in Data Streaming». En: Procedia Computer Science 230, págs. 33-43.

—— (2023c). «Clustering: an R library to facilitate the analysis and comparison of cluster algorithms». En: Progress in Artificial Intelligence 12(1), págs. 33-44.

Laith Mohammad Abualigah, Ahamad Tajudin Khader y Essam Said Hanandeh (2018a). «A new feature selection method to improve the document clustering using particle swarm optimization algorithm». En: Journal of Computational Science 25, págs. 456-466.

—— (2018b). «Hybrid clustering analysis using improved krill herd algorithm». En: Applied Intelligence 48, págs. 4047-4071.

Larry J Eshelman (1991). «The CHC adaptive search algorithm: How to have safe search when engaging in nontraditional genetic recombination». En: *Foundations of genetic algorithms*. (1), págs. 265-283.

Lawrence O. Hall, I. B. Ozyurt y James C. Bezdek (1999). «Clustering with a genetically optimized approach». En: IEEE Trans. Evol. Comput. 3, págs. 103-112.

Leonard Kaufman (1990a). «Monothetic Analysis (Program MONA)». En: Finding groups in data 344, págs. 280-311.

—— (1990b). «Partitioning around medoids (program pam)». En: Finding groups in data 344, págs. 68-125.

Leonard Kaufman y PJ Rousseeuw (1990). «Fuzzy analysis (program FANNY)». En: Finding groups in data: an introduction to cluster analysis, págs. 164-198.

Liyanaarachchi Lekamalage Chamara Kasun, Yan Yang, Guang-Bin Huang y Zhengyou Zhang (2016). «Dimension Reduction With Extreme Learning Machine». En: IEEE Transactions on Image Processing 25(8), págs. 3906-3918.

Long Yin (2020). «Intelligent clustering evaluation of marine equipment manufacturing based on network connection strength». En: Journal of Coastal Research 103(SI), págs. 900-904.

Lotfi Asker Zadeh (1965). «Fuzzy sets». En: Information and control 8(3), págs. 338-353.

—— (1975). «The concept of a linguistic variable and its application to approximate reasoning. Parts I, II, III». En: Information Sciences 8(9), págs. 43-80, 199-249, 301-357.

M Selvakumar y B Sudhakar (2022). «Energy efficient clustering with secure routing protocol using hybrid evolutionary algorithms for mobile adhoc networks». En: Wireless Personal Communications 127(3), págs. 1879-1897.

M.H. Marghny, Rasha Abd El-Aziz y Ahmed Taloba (nov. de 2011). «An Effective Evolutionary Clustering Algorithm: Hepatitis C Case Study». En: International Journal of Computer Applications 34, págs. 123-129.

Madhusmita Das y Rasmita Dash (2020). «Performance analysis of classification techniques for car data set analysis». En: *2020 international conference on communication and signal processing (ICCSP)*, págs. 0549-0553.

Mahamed Omran, Andries Engelbrecht y Ayed Salman (mayo de 2005). «Particle swarm optimization method for image clustering». En: International Journal of Pattern Recognition and Artificial Intelligence 19, págs. 297-321.

Maria Halkidi, Yannis Batistakis y Michalis Vazirgiannis (2001). «On clustering validation techniques». En: Journal of intelligent information systems 17, págs. 107-145.

Mariam Khader y Ghazi Al-Naymat (2019). «An overview of various enhancements of DENCLUE algorithm». En: *Proceedings of the Second International Conference on Data Science, E-Learning and Information Systems*, págs. 1-7.

Marina Meilă (2003). «Comparing clusterings by the variation of information». En: *Learning Theory and Kernel Machines: 16th Annual Conference on Learning Theory and 7th Kernel Workshop, COLT/Kernel 2003, Washington, DC, USA, August 24-27, 2003. Proceedings*, págs. 173-187.

Marina Meilă (2005). «Comparing clusterings: an axiomatic view». En: *Proceedings of the 22nd international conference on Machine learning*, págs. 577-584.

Mario De Luca, Raffaele Mauro, Francesca Russo y Gianluca Dell'Acqua (2011). «Before-after freeway accident analysis using Cluster algorithms». En: Procedia-social and behavioral sciences 20, págs. 723-731.

Martin Maechler et al. (2019). «Finding groups in data: Cluster analysis extended Rousseeuw et al». En: R package version 2(0), págs. 242-248.

Mathilde Caron, Piotr Bojanowski, Armand Joulin y Matthijs Douze (2018). «Deep Clustering for Unsupervised Learning of Visual Features». En: *Computer Vision – ECCV 2018*. (11218), págs. 139-156.

Mayra Z Rodriguez, Cesar H Comin, Dalcimar Casanova, Odemir M Bruno, Diego R Amancio, Luciano da F Costa y Francisco A Rodrigues (2019). «Clustering algorithms: A comparative approach». En: PloS one 14(1), e0210236.

Md Zubair, MD Asif Iqbal, Avijeet Shil, Enamul Haque, Mohammed Moshiul Hoque e Iqbal H Sarker (2021). «An efficient k-means clustering algorithm for analysing covid-19». En: *Hybrid Intelligent Systems: 20th International Conference on Hybrid Intelligent Systems (HIS 2020), December 14-16, 2020*, págs. 422-432.

Mengxuan Zhang, Licheng Jiao, Ronghua Shang, Xiangrong Zhang y Lingling Li (2019). «Unsupervised EA-based fuzzy clustering for image segmentation». En: IEEE Access 8, págs. 8627-8647.

Michael R Berthold y Frank Höppner (2016). «On clustering time series using euclidean distance and pearson correlation». En: arXiv preprint arXiv:1601.02213.

Milton Friedman (1937). «The use of ranks to avoid the assumption of normality implicit in the analysis of variance». En: Journal of the american statistical association 32(200), págs. 675-701.

Minkyu Kim, Suan Lee y Jinho Kim (2020). «A wide & deep learning sharing input data for regression analysis». En: *2020 IEEE International Conference on Big Data and Smart Computing (BigComp)*, págs. 8-12.

Mohammad Norouzi, David J Fleet y Russ R Salakhutdinov (2012). «Hamming distance metric learning». En: Advances in neural information processing systems 25.

Mohammad Rezaei y Pasi Fränti (2016). «Set matching measures for external cluster validity». En: IEEE transactions on knowledge and data engineering 28(8), págs. 2173-2186.

Mohammed H Almannaa, Mohammed Elhenawy y Hesham A Rakha (2019). «A novel supervised clustering algorithm for transportation system applications». En: IEEE transactions on intelligent transportation systems 21(1), págs. 222-232.

Mostafa Mirzaie y Sayyed Majid Mazinani (2017). «Adaptive MCFL: An adaptive multi-clustering algorithm using fuzzy logic in wireless sensor network». En: Computer Communications 111, págs. 56-67.

Mukund Subramaniyan, Anders Skoogh, Azam Sheikh Muhammad, Jon Bokrantz, Björn Johansson y Christoph Roser (2020). «A generic hierarchical clustering approach for detecting bottlenecks in manufacturing». En: Journal of Manufacturing Systems 55, págs. 143-158.

N Raghu Kisore y Ch B Koteswaraiah (2017). «Improving ATM coverage area using density based clustering algorithm and voronoi diagrams». En: Information Sciences 376, págs. 1-20.

Nataliya Boyko, Hanna Komarnytska, Yurii Kryvenchuk y Yuriy Malynovskyy (2019). «Clustering Algorithms for Economic and Psychological Analysis of Human Behavior.» En: *CMiGIN*, págs. 614-626.

Neetu Kushwaha y Millie Pant (2018). «Link based BPSO for feature selection in big data text clustering». En: Future Generation Computer Systems 82, págs. 190-199.

Pawan Lingras, Farhana Haider y Matt Triff (2016). «Granular meta-clustering based on hierarchical, network, and temporal connections». En: Granular Computing 1, págs. 71-92.

Pinki Rani et al. (2017). «A Survey on STING and CLIQUE Grid Based Clustering Methods.» En: International Journal of Advanced Research in Computer Science 8(5).

Purnima Bholowalia y Arvind Kumar (2014). «EBK-means: A clustering technique based on elbow method and k-means in WSN». En: International Journal of Computer Applications 105(9).

R Core Team (2024). *R: A language and environment for statistical computing. R Foundation for Statistical Computing.*

Rakesh Agrawal, Tomasz Imieliński y Arun Swami (1993). «Mining association rules between sets of items in large databases». En: *Proceedings of the 1993 ACM SIGMOD international conference on Management of data*, págs. 207-216.

Ramin Sabbagh y Farhad Ameri (2020). «A framework based on k-means clustering and topic modeling for analyzing unstructured manufacturing capability data». En: Journal of Computing and Information Science in Engineering 20(1), pág. 011005.

Raymond T. Ng y Jiawei Han (2002). «CLARANS: A method for clustering objects for spatial data mining». En: IEEE transactions on knowledge and data engineering 14(5), págs. 1003-1016.

Reza Ghaemi, Md Nasir Sulaiman, Hamidah Ibrahim y Norwati Mustapha (2009). «A survey: clustering ensembles techniques». En: International Journal of Computer and Information Engineering 3(2), págs. 365-374.

Rich Caruana, Mohamed Elhawary, Nam Nguyen y Casey Smith (2006). «Meta clustering». En: *Sixth International Conference on Data Mining (ICDM'06)*, págs. 107-118.

Romana Capor-Hrosik, Eva Tuba, Edin Dolicanin, Raka Jovanovic y Milan Tuba (jul. de 2019). «Brain Image Segmentation Based on Firefly Algorithm Combined with K-means Clustering». En: Studies in Informatics and Control 28.

Ryota Suzuki e Hidetoshi Shimodaira (2006). «Pvclust: an R package for assessing the uncertainty in hierarchical clustering». En: Bioinformatics 22(12), págs. 1540-1542.

S Mythili y E Madhiya (2014). «An analysis on clustering algorithms in data mining». En: International Journal of Computer Science and Mobile Computing 3(1), págs. 334-340.

Sabine Landau e Irina Chis Ster (2010). «Cluster Analysis: Overview». En: International Encyclopedia of Education, págs. 72-83.

Sanghamitra Bandyopadhyay y Ujjwal Maulik (2002). «Genetic clustering for automatic evolution of clusters and application to image classification». En: Pattern Recognition 35(6), págs. 1197-1208.

Sarah Shafqat, Saira Kishwer, Raihan Ur Rasool, Junaid Qadir, Tehmina Amjad y Hafiz Farooq Ahmad (2020). «Big data analytics enhanced healthcare systems: a review». En: The Journal of Supercomputing 76, págs. 1754-1799.

Satya Chaitanya Sripada y M Sreenivasa Rao (2011). «Comparison of purity and entropy of k-means clustering and fuzzy c means clustering». En: Indian journal of computer science and engineering 2(3), págs. 343-346.

Satyasai Jagannath Nanda y Ganapati Panda (2014). «A survey on nature inspired metaheuristic algorithms for partitional clustering». En: Swarm and Evolutionary computation 16, págs. 1-18.

Saumya Singh y Smriti Srivastava (2020). «Review of clustering techniques in control system: Review of clustering techniques in control system». En: Procedia Computer Science 173, págs. 272-280.

Seref Sagiroglu y Duygu Sinanc (2013). «Big data: A review». En: *2013 international conference on collaboration technologies and systems (CTS)*. IEEE, págs. 42-47.

Shahina Anjum, Sunil Kumar Yadav y Seema Yadav (2024). «Stream Data Model and Architecture». En: *Data Analytics and Machine Learning: Navigating the Big Data Landscape*, págs. 81-104.

Silke Wagner y Dorothea Wagner (ene. de 2007). «Comparing Clusterings - An Overview». En: Technical Report 2006-04.

Sriparna Saha, Ranjita Das y Dr. Partha Pakray (sep. de 2018). «Aggregation of multi-objective fuzzy symmetry-based clustering techniques for improving gene and cancer classification». En: Soft Computing 22.

Sriparna Saha y Sanghamitra Bandyopadhyay (2009). «A validity index based on connectivity». En: *2009 Seventh International Conference on Advances in Pattern Recognition*, págs. 91-94.

Sture Holm (1979). «A simple sequentially rejective multiple test procedure». En: Scandinavian journal of statistics, págs. 65-70.

Sudipto Guha, Rajeev Rastogi y Kyuseok Shim (1998). «CURE: An efficient clustering algorithm for large databases». En: ACM Sigmod record 27(2), págs. 73-84.

Sujoy Bag, Sri Krishna Kumar y Manoj Kumar Tiwari (2019). «An efficient recommendation generation using relevant Jaccard similarity». En: Information Sciences 483, págs. 53-64.

Teng Li, Amin Rezaeipanah y ElSayed M Tag El Din (2022). «An ensemble agglomerative hierarchical clustering algorithm based on clusters clustering technique and the novel similarity measurement». En: Journal of King Saud University-Computer and Information Sciences 34(6), págs. 3828-3842.

Tian Zhang, Raghu Ramakrishnan y Miron Livny (1996). «BIRCH: an efficient data clustering method for very large databases». En: ACM sigmod record 25(2), págs. 103-114.

Tien Anh Tran (2020). «Effect of ship loading on marine diesel engine fuel consumption for bulk carriers based on the fuzzy clustering method». En: Ocean Engineering 207, pág. 107383.

Tomáš Löster (2016). «Determining the optimal number of clusters in cluster analysis». En: Proceedings of the 10th International Days of Statistics and Economics, Prague, Czech Republic 8(10).

Tossapon Boongoen y Natthakan Iam-On (2018). «Cluster ensembles: A survey of approaches with recent extensions and applications». En: Computer Science Review 28, págs. 1-25.

Ujjwal Maulik y Sanghamitra Bandyopadhyay (2000). «Genetic algorithm-based clustering technique». En: Pattern recognition 33(9), págs. 1455-1465.

Ulrich Bodenhofer, Andreas Kothmeier y Sepp Hochreiter (2011). «APCluster: an R package for affinity propagation clustering». En: Bioinformatics 27(17), págs. 2463-2464.

Usama Fayyad (1997). «Knowledge discovery in databases: An overview». En: *International Conference on Inductive Logic Programming*, págs. 1-16.

Usama Fayyad, Gregory Piatetsky-Shapiro y Padhraic Smyth (1996). «From data mining to knowledge discovery in databases». En: AI magazine 17(3), págs. 37-37.

Usama Fayyad y Paul Stolorz (1997). «Data mining and KDD: Promise and challenges». En: Future Generation Computer Systems 13(2), págs. 99-115.

V Priya y K Umamaheswari (2020). «Aspect-based summarisation using distributed clustering and single-objective optimisation». En: Journal of Information Science 46(2), págs. 176-190.

Wei Song, Yingying Qiao, Soon Cheol Park y Xuezhong Qian (2015). «A Hybrid Evolutionary Computation Approach with Its Application for Optimizing Text Document Clustering». En: Expert Systems with Applications 42(5), págs. 2517-2524.

Xin-She Yang (2010). *Nature-inspired metaheuristic algorithms*.

Yadi Wang, Zefeng Zhang y Yinghao Lin (2021). «Multi-cluster feature selection based on isometric mapping». En: IEEE/CAA Journal of Automatica Sinica 9(3), págs. 570-572.

Yanfang Han y Pengfei Shi (ene. de 2007). «An improved ant colony algorithm for fuzzy clustering in image segmentation». En: Neurocomputing 70, págs. 665-671.

Yang Yang, Zhigang Ma, Yi Yang, Feiping Nie y Heng Tao Shen (2014). «Multitask spectral clustering by exploring intertask correlation». En: IEEE transactions on cybernetics 45(5), págs. 1083-1094.

Yiming Mao, Deborah Simon Mwakapesa, Kaibin Xu, Chen Lei, Youcun Liu y Maosheng Zhang (2021). «Comparison of wave-cluster and DBSCAN algorithms for landslide susceptibility assessment». En: Environmental Earth Sciences 80, págs. 1-14.

Ying Ju, Songming Zhang, Ningxiang Ding, Xiangxiang Zeng y Xingyi Zhang (sep. de 2016). «Complex Network Clustering by a Multi-objective Evolutionary Algorithm Based on Decomposition and Membrane Structure». En: Scientific reports 6, pág. 33870.

Yonglai Zhang y Yaojian Zhou (2019). «Review of clustering algorithms». En: Journal of Computer Applications 39(7), pág. 1869.

Yusak Tanoto, Navid Haghdadi, Anna Bruce y Iain MacGill (2020). «Clustering based assessment of cost, security and environmental tradeoffs with possible future electricity generation portfolios». En: Applied Energy 270, pág. 115219.

Zhexue Huang (1997). «A fast clustering algorithm to cluster very large categorical data sets in data mining.» En: Dmkd 3(8), págs. 34-39.