Aarón Casado Monge[1], Juan José Cuadrado Gallego[1]

University of Alcala, Polytechnic School, Computer Science Department, Scientific and Technological Campus, Politechnic Building. Office: O243, 28805, Alcala de Henares, Madrid, Spain

Abstract

Clusterización (qué es y para qué sirve) - Técnicas (para qué sirven) - Paquetes

Keywords: Data Mining, Clustering, R

1. Introducción

Desde finales del sigo XX se ha considerado que vivimos en la "era de la información", una etapa caracterizada por el incremento, desarrollo y propagación de emergentes tecnologías de la información y comunicación que han permitido al ser humano romper las barreras de la distancia, el tiempo y el lugar a la hora de comunicarse y compartir información; actividades que han sido decisivas en nuestra historia [1]. Sin embargo, la era en la que realmente vivimos es la "era de los datos", donde cada día se generan más de veiticinco mil petabytes[1] de datos provenientes de comercios, ciencias, Internet y casi cualquier actividad del día a día [2] que acaban volcados en redes de ordenadores, sitios web, bases de datos y otros medios de almacenaje.

Esta explosión de datos, a la que se ha denominado Big Data, se debe al alto grado de computarización de la sociedad y el avance de herramientas de recolección y almacenamiento de datos. Negocios en todo el mundo generan grandes cantidades de datos derivados de transacciones, stock de productos, platillas de empleados, etc. Las ramas de la ciencia producen datos de manera constante frutos de experimentos, observaciones, recogida de muestras, etc. Y más recientemente, Internet y las redes sociales han sido las principales responsables del aumento excesivo de datos, siendo usadas por millones de personas simultáneamente.

Y, aunque esto ha supuesto una considerable mejora para la humanidad pues la información nunca había sido tan accesible, también ha traido consecuencias negativas y problemas como el almacenamiento y organización de los datos, datos no estructurados que entorpecen su acceso y procesamiento, dificultades a la hora de analizar los datos apropiadamente pudiendo generar desinformación y complicaciones para mostrar los resultados de forma apropiada y aplicarlos de manera eficiente y útil en el mundo real [3].

Como resultado, ha surgido una nueva ciencia que se ha posicionado rápidamente como una de las disciplinas más influyentes de la actualidad: Data Science (Ciencia de los Datos), que debido a su reciente aparición, carece de una definición consensuada, pero podríamos concretarla como "*Ciencia que usa Estadística, Inteligencia Artificial, Programación y Bases de Datos para posibilitar la extracción de conocimiento a partir de datos*" [4]. A su vez, dentro de esta ciencia se han desarrollado otras tres ramas: Data Warehousing, Data Mining y Visualization; cada una de ellas enfocada a resolver o afrontar uno o varios de los problemas mencionados previamente: organización y agrupación de datos, análisis de los mismos y presentación de los resultados, respectivamente.

De entre estas nuevas disciplinas, Data Mining es la que se centra en el procesamiento de los datos, procedimiento por el cual se obtiene la información, y se podría definir como "*Proceso de descubrimiento de patrones interesantes y conocimiento a partir de grandes volúmenes de datos*" [5], donde dentro de la misma podemos encontrar diferentes técnicas para encontrar patrones y relaciones, y dependiendo de cuál se aplique se puede obtener un resultado to-

Email addresses: aaron.casado@uah.es (Aarón Casado Monge), jcg@uah.es (Juan José Cuadrado Gallego)

[1]Un Petabyte es una unidad de información o almacienamiento de datos equivalente a un cuadrillon de bytes, mil terabytes o un millón de gigabytes. En este caso, es el equivalente a 2.5 trillones de bytes.

talmente diferente incluso con el mismo conjunto de datos, por lo que es fundamental emplear la técnica apropiada en función del del objetivo a conseguir, los datos con los que se pretende trabajar y el ámbito de aplicación.

Y en algunos de los campos más importantes del mundo moderno como la analítica de negocios, el reconocimiento de imágenes, las búsquedas web, seguridad, biología y ciencias de la salud; existen dificultades a la hora de clasificar o agrupar ciertos datos porque estos no disponen de una etiqueta o valor conocido por el que se pueda hacerlo, pues este no existe o no ha sido definido. Para poder afrontar este problema se utiliza la técnica de clustering, que permite exactamente generar valores y etiquetas para un conjunto de datos realizando agrupaciones denominadas *clusters* o grupos, donde los datos de un mismo cluster sean muy similares entre ellos y a su vez tengan diferencias claras con datos de otros clusters, permitiendo que cada grupo resultante puede ser etiquetado y tratado como una clase propia.

De esta manera, el objetivo de este Trabajo de Fin de Grado (TFG) es realizar un estudio sobre clustering, exponiendo de manera teórica qué es este método y qué tipo de utilidades tiene, así como el desarrollo de las diferentes técnicas que existen y las aplicaciones que estas ofrecen. Posteriormente se explorará este método dentro de Bioinformática, un campo centrado en desarrollar técnicas y programas software para analizar datos biológicos, viendo qué aporta a dicha disciplina y cuáles de las técnicas expuestas previamente se emplean y por qué.

Una vez realizado el marco teórico previo, se pretende hacer una parte práctica los diferentes paquetes que ofrecen técnicas de clustering tanto de carácter general como dentro de Bioinformática para el lenguaje de programación R [6], uno de los más relevantes dentro de Data Science.

2. Clustering

Clustering o cluster analysis, en español agrupamiento o análisis de grupos y adaptado al término clusterización, es un método de Data Mining que se basa en el Aprendizaje Automático (Machine Learning), una rama de la Inteligencia Artificial (Artificial Intelligence) que pretende desarrollar sistemas que resuelvan problemas basándose en los resultados de experiencias previas, aprendiendo de sus errores. Concretamente, se fundamenta en uno de los métodos de aprendizaje explorados en esta disciplina: Aprendizaje No Supervisado, enfocado en el desarrollo y descubrimiento de nuevos conocimientos. Este, a diferencia de otros métodos de aprendizaje, no dispone de conocimiento previo sobre lo que poder aprender, por lo que su objetivo principal es discernir patrones y relaciones entre los datos para poder separarlos [7].

En esencia, el proceso de clustering es el mismo, hasta el punto en que básicamente podríamos considerarlos sinónimos, puesto que clusterización se aplica principalmente sobre conjuntos de datos que se desean organizar en diferentes grupos pero los valores que delimitan cada cluster son desconocidos y por lo tanto se definen en el propio proceso de clasificación.

De una manera más técnica, podríamos decir que clustering busca definir para una determinada característica[2] o Suceso Elemental[3] (SE), un conjunto de grupos de observaciones (suceso) [4] con valores cercanos; donde los clusters permiten, dados los diferentes sucesos elementales que configuran un suceso, asignar cada SE al mismo cluster [8].

Para poder decidir si dos datos deben ser agrupados en el mismo cluster o por el contrario, separados, es necesario introducir los conceptos de similaridad y disparidad; cuanto más similares sean los datos, más tendrán en común y por lo tanto es más probable que terminen bajo el mismo cluster, y por otra parte, si los datos son diferentes entre sí, serán separados en clusters diferentes. Este criterio se basa en la proximidad de dichos objetos. Si midiéramos la similitud entre dos objetos i y j, esta devolvería 0 si ambos fueran totalmente distintos, y cuanto más elevado fuera el valor, más semejantes serían, siendo 1 el valor más alto, indicando que ambos datos son idénticos. De la misma manera, medir la disparidad entre los objetos daría resultados opuestos, con un 0 indicando que son idénticos y con un 1 que no tienen nada en común.

Esta forma de calcular la proximidad no solo ayuda a la hora de agrupar los datos en clusters y formar subconjuntos a partir de los resultados, sino que con este proceso también somos capaces de detectar outliers, datos anómalos que pueden ser

[2] Dicho de una cualidad que da carácter o sirve para distinguir a algo o alguien de sus semejante.

[3] Cada uno de los resultados más simples que se pueen obtener de la realización de un experimento. Obtener el número 3 al tirar un dado.

[4] Se denomina así al subconjunto total de resultados posibles al realizar un experimento. Obtener un 3 o sacar par al tirar un dado.

datos erróneos procedentes de errores de medida o fallos que deben ser eliminados o, datos correctos sumamente importantes que son diferentes al esto y deben ser analizados detenidamente. A la hora de agrupar los datos, puede que queden varios objetos sueltos sin pertenecer a ningún cluster, esos son los datos anómalos.

Además, clustering también sirve como primera aproximación a la hora de procesar los datos, porque aunque de por sí, este proceso puede aportar información útil agrupando datos similares y clasificándolos, permitiendo un análisis de los resultados que puede dar lugar al descubrimiento de nuevos conocimientos, también se usa como método de preprocesamiento de datos para otros algoritmos de Data Mining que trabajarán sobre las clusters generados y los atributos seleccionados como criterio a la hora de crearlos, pues estos se pueden considerar clases nuevas de objetos.

Es gracias a estas características que clusterización es empleada en muchos ámbitos del mundo actual, siendo la técnica de Aprendizaje No Supervisado más extendida. Campos como la analítica de negocios, el reconocimiento de imágenes, las búsquedas web, seguridad, biología y ciencias de la salud hacen uso habitual de esta técnica para clasificar tipos de consumidores que comparten preferencias, aunar bajo un mismo subconjunto muchas formas diferentes de escribir el mismo caracter para facilitar el reconocimiento de textos escritos a mano, agrupar resultados similares de una consulta en internet y mostrar los más relevantes dentro de cada grupo o el estudio de la taxonomía[5] de las especies. También se puede hacer uso de su capacidad para detectar datos anómalos con la finalidad de revelar posibles fraudes o transacciones financieras sospechosas, incluso ayudar en la disminución de crímenes analizando los resultados obtenidos tras clusterizar datos de detenciones y delitos; incluso aplicada a datos geofísicos[6] para interpretarlos y obtener resultados significativos. Asimismo, se utiliza a la hora de comparar comunidades en redes sociales permitiendo recomendar a los usuarios contenido de su agrado, o dentro del mundo sanitario, ayudando en la identificación y control de diversos tipos de enfermedades e incluso puede usarse como apoyo en la gestión de edificios públicos como bibliotecas, agrupando a los lectores por sus preferencias de manera similar a los consumidores de un negocio [9–13].

Comprimiendo toda esta información, podemos concretar los tres principales motivos por los que clustering es empleado: clasificación natural tanto de individuos, como de objetos, etc.;, compresión de información para mayor facilidad a la hora de mostrar resultados y generación de una estructura base para el posterior tratamiento de los datos [14].

La utilidad de este método y la flexibilidad que ofrece con las diversas técnicas y formas de aplicarlo de las que dispone e es también parte fundamental de que sea tan comúnmente utilizado y con objetivos tan dispares en gran parte de las áreas del conocimiento. Sin embargo, también es un método frágil, pues depende en gran medida de los datos con los que se trabaje y el criterio escogido para determinar si dos objetos deben agruparse bajo el mismo cluster o separarlos, Por lo que es necesario seguir una serie de pasos a la hora de realizar un análisis de grupos de manera correcta [15]:

1. Primero, hay que seleccionar cuidadosamente los datos con los que se va a trabajar, puesto que tanto el tipo de dato como la cantidad de los mismos influyen directamente en los resultados. Si se utilizan demasiados datos el procesamiento puede tener un coste computacional alto y es difícil ofrecer una visualización elegante de los resultados. Pero si se escogen pocos datos se pierde información útil y puede dar lugar a equivocaciones. Es por ello que este paso suele realizarlo un experto en el sector o con su ayuda.
2. Segundo, se debe elegir la forma en la que se va a calcular la similaridad entre los diferentes datos, que analizaremos más adelante cuando analicemos las diferentes técnicas de clustering.
3. Tercero, debe definirse el criterio con el que se va a tomar la decisión de agrupar en el mismo cluster dos datos. Es decir, escoger el umbral de similaridad a partir del cual dos datos pasan a formar parte del mismo cluster y qué característica o conjunto de ellas van a ser utilizadas para medir la similitud. La elección del umbral suele ser complicada, y es necesaria mucha experiencia o varias iteraciones de ensayo y error para encontrar un valor correcto.
4. Cuarto, hay que optar por uno de los diversos algoritmos de clusterización que existen,

[5] Ciencia que trata de los principios, métodos y fines de la clasificación. Se aplica en particular, dentro de la biología, para la ordenación jerarquizada y sistemática, con sus nombres, de los grupos de animales y de vegetales.

[6] La Geofísica es la ciencia que estudia la Tierra desde el punto de vista de la física.

pues los resultados pueden cambiar considerablemente al variar la estrategia con la que se realiza la agrupación. Estos se verán a continuación.

5. Por último, una vez obtenidos los resultados hay que validarlos e interpretarlos. La resolución de este paso depende del objetivo inicial por el que se haya decidido realizar el análisis.

Como puede verse, tanto el paso inicial como el último requieren de la ayuda de personas cualificadas si pretenden realizarse correctamente, y es en los pasos intermedios donde la intervención de computadoras y programas informáticos son más útiles y están más desarrollados, pues permiten aligerar considerablemente la carga de trabajo y como consecuencia, se puede analizar una mayor cantidad de datos. Pero no deja de ser una tarea complicada que ha ocasionado una gran demanda de expertos en Data Mining y Data Science en diversas áreas de trabajo durante los últimos años.

La utilidad que aporta clustering permitiendo crear grupos con datos similares entre sí y dispares con respecto a otros grupos con el fin de clasificarlos y las ventajas que esto presenta queda reflejada en la cantidad de usos que recibe, por lo que vamos a profundizar en las diferentes técnicas que son empleadas para la generación de los clusters y su funcionamiento que lo hacen tan importante.

3. Técnicas de clustering

Existe una gran variedad de técnicas de clustering que cumplen con el propósito de clasificar un conjunto de datos, pero no todas son iguales. Existen una serie de requisitos típicos que debe cumplir un algoritmo de clusterización para satisfacer las expectativas y realizar un buen análisis. Sin embargo, cumplir todos estos requerimientos de forma simultánea es una tarea casi imposible, por lo que analizar cuáles ofrece cada algoritmo nos ayuda a identificar sus puntos fuertes y débiles y por ende, facilitar la elección del algoritmo apropiado para cada ocasión. Veamos cuáles son dichos requisitos y cómo podemos comparar algoritmos.

*3.1. **Requisitos de un algoritmo de clustering***

[16] Sobre el papel, analizar datos no debería suponer un problema; pero la realidad es que existen trabas que complican este procedimiento tales como la cantidad de datos a analizar o el tipo de datos. Algunos de los algoritmos de clustering iniciales no tenían en cuenta este tipo de problemas, porque no existían en su momento, pero ahora es necesario afrontarlos para que el análisis de los datos sea eficiente y produzca resultados completos y veraces.

A continuación, se exponen los requisitos típicos que se exigen dentro de Data Mining a un algoritmo de clustering, así como debilidades que presentan algunos de los existentes.

- **Escalabilidad**: Muchos algoritmos de clusterización trabajan bien con conjuntos de datos pequeños que contienen solo unos cuantos cientos de objetos; sin embargo, la gran mayoría de bases de datos contienen millones y miles de millones de datos. Analizar solo un pequeño porcentaje de esos datos puede resultar en conocimiento parcial, siendo necesario que sean escalables para resultados óptimos.

- **Habilidad para lidiar con diferentes tipos de atributos**[7]: Coloquialmente se suele entiende por atributo un dato[8] cuantitativo, es decir, un número. Sin embargo, existen otros tipos de datos: nominales[9], binarios[10], ordinales[11], imágenes, gráficos, documentos y mezclas de varios de estos tipos. Por lo que un algoritmo de clustering debe estar preparado para lidiar con todos o la gran mayoría de ellos.

- **Descubrimiento de clusters con formas arbitrarias**: Gran parte de los algoritmos utilizan formas de calcular la similaridad entre los datos basadas en la distancia entre los mismos, lo que suele dar lugar a clusters con formas redondas o esféricas. En determinadas ocasiones esto puede resultar contraproducente y erróneo, por lo que también es preciso que los algoritmos sean capaces de identificar diferentes formas de clusters, no solamente formas circulares o geométricas.

[7] Dicho de una característica cualitativa de un individuo o dato usada para distinguirlo de una variable o característica cuantitativa.

[8] Valor obtenido para una característica en una observación.

[9] Dicho de un dato cualitativo que proporciona suficiente información para nombrar a una característica y poder diferenciarla de otra.

[10] Datos que solo pueden tomar los valores cero y uno (0, 1).

[11] Datos cualitativos que proporcionan suficiente información para ordenar las observaciones.

- **Requirir información al usuario**: En algunos algoritmos de clustering es común pedir cierto tipo de información al usuario a la hora de trabajar con los datos, como el número de clusters que se quieren generar. Este tipo de práctica puede ocasionar problemas dada la complejidad de esta decisión, sobre todo en escenarios en los que los datos cuentan con múltiples atributos. Por lo que es recomendable que los algoritmos eviten este comportamiento tanto para aliviar a los usuarios como para no comprometer la calidad a la hora de formar clusters, pues el algoritmo se adaptaría al valor introducido por el usuario, aunque fuese erróneo, generando también, un resultado errado.

- **Capacidad para trabajar con ruido en los datos**: Aunque es común tratar de antemano los datos para eliminar cualquier tipo de dato no deseado, recordemos que esta técnica también se aplica como método de preprocesamiento para otros algoritmos de Data Mining, por lo que es necesario que no sean sensibles hacia estas impurezas, puesto que podría resultar en una clasificación pobre.

- **Clusterización incremental e insensibilidad al orden de entrada**: Cuando se aplica clustering en entornos en movimiento constante como el de los negocios, es común que se introduzcan datos nuevos junto a los ya existentes. Ciertos algoritmos no son compatibles con este incremento de información, y es necesario volver a iniciar un nuevo proceso de clusterización. Mientras que otros que sí toleran este cambio, resulta que son sensibles respecto al orden con el que se han introducido los datos y pueden generar agrupaciones erróneas. Estos problemas han dado lugar a la necesidad de algoritmos que puedan lidiar con la incorporación de nuevos datos y que no se vean afectados por el orden en el que estos son introducidos.

- **Capacidad de Clusterizar datos con muchos atributos**: Aunque de manera común se suele trabajar con datos que no contienen mucho más que una decena de atributos, hay ciertos tipos de datos como documentos, en los que cada palabra clave puede ser considerada un atributo, dando lugar a cientos de atributos por cada dato, haciendo complicado trabajar con ellos y también clasificarlos, por lo que se precisa de algoritmos que puedan lograr un buen resultado con este tipo de datos.

- **Clustering basado en restricciones**: En el mundo real podemos encontrar ciertas barreras a la hora de aplicar un algoritmo de clustering, en los que, por ejemplo, sí se tenga que tener en cuenta un límite a la cantidad de cluster que se pueden generar o de decidir si dos datos o clusters deban unirse. Aunque es complicado realizar este tipo de agrupaciones, es indispensable que los algoritmos obtengan resultados decentes dadas estas limitaciones.

- **Interpretabilidad y usabilidad**: De nada sirve clasificar los datos y cumplir con los requisitos previos si no se puede interpretar el resultado. Los algoritmos, por lo tanto, deben ofrecer conclusiones comprensibles, coherentes y que puedan usarse, sometiéndose a ciertas limitaciones semánticas y visuales con la finalidad de acomodar el resultado al objetivo inicial por el que se ha llevado a cabo la clusterización.

3.2. Criterios de comparación

[16] Cada algoritmo opera de forma distinta en diferentes aspectos como el criterio de división, la separación de clusters, el cálculo de la similaridad y el espacio de clusterización. Observando los pasos a seguir y la toma de decisiones de un algoritmo, podemos compararlo con otro y escoger cuál es más adecuado para el problema a afrontar.

- **Criterio de división**: Principalmente existen dos tipos de métodos respecto a este criterio, aquellos que fraccionan todos los datos para que no exista ningún tipo de orden o diferencia de nivel entre ellos, y los que, por el lado opuesto, dividen los datos de manera jerárquica, permitiendo formar clusters en diferentes niveles semánticos. Cada uno aporta una utilidad distinta, con el primer tipo de criterio podemos abordar problemas como la clasificación de clientes o lectores, conjuntos de datos en los todos reciben el mismo trato; mientras que con criterio de partición jerárquico podemos organizar documentos en varios campos como “deportes”, “política” o “comida”, y dentro de cada campo tener subconjuntos como “postres”, “pasta” o “carne”.

- **Separación de clusters**: Otro criterio en el que hay dos enfoques: si un dato forma parte de un cluster, este no puede formar parte de otro a no ser que ambos clusters se unan; o permitir que un dato pueda estar en dos clusters a la vez, un concepto denominado "Fuzzy clustering". La primera aproximación ofrece un enfoque más determinista que es quizás el más aplicado entre ambos métodos, pero existen disciplinas en las que permitir que un dato forme parte de más de cluster puede ser realmente útil y que cada vez recibe más usos [17].

- **Cálculo de similaridad**: Una de las pautas que más determina un algoritmo de clusterización es la forma en la que calculan la similaridad entre los datos, y que exploraremos detalladamente más adelante. Gran parte de los algoritmos utilizan la distancia que hay entre dos objetos para calcular su similaridad aplicando métodos como la distancia Euclídea o espacios vectoriales. Esta tipo de cálculo se ve beneficiado de métodos de optimización, pero suele generar clusters con formas esférica o cilíndricas, por lo que cada vez más se utilizan algoritmos que miden la similaridad entre objetos mediante la densidad o contigüidad del espacio para formar clusters, lo cual permite descubrir grupos con formas arbitrarias.

- **Espacio de clusterización**: Muchos algoritmos de clustering observan todo el espacio de datos para buscar grupos, que es un método totalmente factible y válido para datos con pocos atributos. Sin embargo, cuando se tratan de procesar datos con gran cantidad de atributos, es mejor centrarse en pequeñas partes del total, en subconjuntos de espacio y buscar clusters dentro de ellos de forma que se pueda obtener información valiosa a un menor coste computacional.

3.3. Clasificación general de técnicas

[16] Una vez que hemos visto los requisitos de los algoritmos de clustering y los criterios para compararlos entre sí, podemos agrupar la mayor parte de las técnicas bajo cuatro grandes grupos basados principalmente en el cálculo de la similaridad y el criterio de división. Hablaremos brevemente sobre estos a continuación y posteriormente, precisaremos de manera individual sobre cada uno de ellos en las secciones 4, 5, 6 y 7, señalando los principales algoritmos dentro de cada una de ellos.

- **Métodos basados en particiones**: Este tipo de métodos dividen todos los objetos de un conjunto de datos en clusters en los que debe haber al menos un objeto; o dicho de una forma más técnica, teniendo n objetos, se generarán k particiones siendo $k \leq n$.

  Normalmente las técnicas de este grupo generan clusters exclusivos, donde un datos solo puede pertenecer a un cluster, pero cada vez se desarrollan más algoritmos de tipo fuzzy que relajan la agrupación. Asimismo, muchos de estos métodos calculan la distancia entre los objetos como medida de similaridad y generan de manera inicial k particiones para posteriormente, de forma iterativa, tratar de reagrupar los datos y clusters para mejorar la clasificación, basándose en que los datos dentro del mismo cluster estén realmente cerca entre sí, y que con los datos de otros clusters se encuentren lejos.

  Estos métodos sirven tanto para búsquedas en todo el espacio, como clusterización de espacios reducidos, pero cuando la cantidad de atributos es muy alta, es preferible aplicarlos en subconjuntos, pues es complicado optimizar computacionalmente este tipo de algoritmos a la hora de generar las particiones iniciales, y para lidiar con este inconveniente, se suelen usar métodos heurísticos como *k-means* que mejoran la calidad de clustering progresivamente. Sin embargo, como se ha mencionado previamente, esta aproximación de clustering genera grupos con formas esféricas y circulares, por lo que uno de los principales focos de mejora de estos algoritmos es evitar ese tipo de comportamiento cuando se trabaja con grandes conjuntos de datos.

- **Métodos jerárquicos**: Las técnicas de este grupo crean, como su propio nombre indica, una separación jerárquica de los datos. De forma diferente a los métodos basados en particiones donde todos los datos están en el mismo nivel, aquí se busca generar varios rangos para la clusterización.

  Existen dos enfoques principales dependiendo de cómo se generen los clusters: *aglomerativo* y *divisivo*. Las técnicas aglomerativas parten de la separación de todos los objetos en clusters individuales y progresivamente los va uniendo hasta que solo queda uno; mientras que las

técnicas divisivas parten de un solo cluster inicial en el que se encuentran todos los objetos y va formando clusters más pequeños hasta que cada objeto es en sí un cluster. Es por eso que también se les conoce por métodos de "abajo a arriba" y de "arriba a abajo" respectivamente.

Esta técnicas pueden utilizar distancia, densidad y continuidad como medida de calcular la similaridad entre los datos y son útiles tanto en búsquedas de espacios completos como de subconjuntos.

- **Métodos basados en densidad**: Este tipo de métodos surgieron como solución a los métodos basados en particiones y su tendencia a encontrar únicamente clusters con formas esféricas o circulares, puesto que las técnicas que utilizan la densidad del espacio a la hora de formar clusters permiten descubrir clusters con formas más arbitrarias.

  El concepto general de estos métodos consiste en agrandar un cluster mientras la densidad en el "vecindario" este por encima de un umbral. La densidad se calcula por la cantidad de objetos que hay en el espacio de datos y el vecindario hace referencia a todos los datos que se encuentran a cierta distancia de un objeto determinado.

  El concepto general es que para cada objeto dentro de un cluster, en su vecindario de radio r de distancia debe haber un mínimo de x objetos. Este tipo de aproximación resulta muy conveniente a la hora de detectar datos anómalos.

  Los métodos basados en densidad pueden clasificar los datos en clusters exclusivos o de manera jerárquica, y aunque normalmente consideran que un objeto solo puede pertenecer a un cluster, también existen técnicas fuzzy clustering. Asimismo, se pueden aplicar tanto en espacios completos como en regiones puntuales.

- **Métodos basados en cuadrícula o rejilla**: Estos métodos dividen el espacio de objetos en una cantidad determinada de celdas que forman una estructura de cuadrícula o rejilla, de ahí su nombre. Todas las operaciones con los datos se realizan dentro de la rejilla, que permite un procesamiento mucho más rápido, pues no depende de la cantidad de datos, sino de la de celdas existentes.

  Esto permite que puedan ser integrados con otros métodos de clustering de otros grupos como jerárquicos y basados en densidad para mejorar su eficiencia.

En la tabla 1 se han resumido estos grupos y las principales características de los mismos. Cabe mencionar que esta clasificación no es perfecta, pues existen algoritmos que integran conceptos de diferentes métodos.

A continuación, exploraremos todos estos grupos y las principales técnicas empleadas en cada uno de ellos.

4. Métodos basados en particiones

Los métodos pertenecientes a este grupo son quizás los más simples y básicos de clustering. Estos organizan los datos de un conjunto en diversos clusters exclusivos, y es común que soliciten al usuario el número de grupos o clusters finales que debe haber, pues sirve como punto de partida para estos métodos. Esto permite simplificar su funcionamiento, pero a su vez, supone el incumplimiento de uno de los requisitos de los que se ha hablado previamente en la sección 3.1.

De forma técnica, dado un conjunto de datos D con n objetos y siendo k el número de clusters a formar, un algoritmo basado en particiones organizará los objetos en k divisiones, con $k \leq n$, donde cada una de ellas representa un cluster.

Los clusters son generados intentando optimizar el criterio de similaridad, siendo uno de los más utilizados el del cálculo de proximidad, donde se intentan agrupar bajo el mismo clusters aquellos objetos que estén cerca en el espacio de datos y que a su vez, hacen que todos los datos de un cluster estén alejados de los objetos de otro cluster, asegurando de esta manera un alto grado de semejanza entre los del mismo cluster y diferencia con los de otros clusters.

A continuación se exponen los dos algoritmos más utilizados de este grupo y de los más conocidos: k-media (k-means) 4.1 y k-medoids 4.2.

4.1. ***k-Means***

El objetivo de los algoritmos basados en particiones es, dado un conjunto de datos D con n objetos en un espacio Euclídeo[12], formar k clusters,

[12] Espacio geométrico donde se satisfacen los axiomas de Euclides.

Método	Características generales
Basados en particiones	- Encuentran clusters de forma esférica mutuamente exclusivos - Utilizan distancia/proximidad - Usan la media o medoid para representar el centro del cluster - Efectivos en conjuntos de datos pequeños y medianos
Jerárquicos	- Clasifican en múltiples niveles - No pueden deshacer agrupaciones o divisiones erróneas - Pueden incorporar otras técnicas como microclustering y tener en cuenta vínculos entre los objetos
Basados en densidad	- Pueden encontrar clusters con formas arbitrarias - Los clusters son regiones con gran densidad de objetos separados por zonas con poca densidad - La densidad queda definida por un mínimo de objetos cercanos dentro del vecindario - Sirve para detectar datos anómalos
Basados en rejilla	- Utiliza una estructura de rejilla o cuadrícula - La velocidad de procesamiento es alta, pues no influye el número de objetos

Table 1: Resumen de métodos de clusterización

$C_1, ..., C_k$, donde cada cluster es un subconjunto del conjunto de datos $C_i \subset D$ y la intersección entre clusters es nula $C_i \cap C_j = \emptyset$ para $i \leq j$ y $j \leq k$. Sin embargo, estos requieren de una gran capacidad computacional, y es por ello que algoritmos como k-Means son tan utilizados.

k-Means representa cada cluster con su "centroide", que es básicamente el punto central del cluster, y se calcula como la media de los puntos que lo forman. Para un conjunto de datos con dos atributos cuantitativos, la fórmula para calcular el centroide c_i es la siguiente:

$$c_i = \left(\frac{\sum_{i=1}^{n} x_i}{n}; \frac{\sum_{j=1}^{m} y_j}{m} \right). \tag{1}$$

Asimismo, otro de los métodos más comunes para medir la distancia entre puntos y clusters, y que también utiliza k-Means, es la distancia Euclídea. La fórmula de la distancia Euclídea entre dos puntos cualesquiera $dist(p, q)$ de un espacio Euclídeo p y q, con coordenadas cartesianas[13] (x_1, y_1) y (x_2, y_2) es como sigue:

$$dist\,(p, q) = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}. \tag{2}$$

Y para comprobar la calidad de un cluster, y garantizar que los objetos del un mismo sean similares entre ellos, pero diferentes con objetos de otros clusters, los métodos basados en particiones suelen utilizar la *varianza* dentro del cluster, que es la suma del error cuadrático[14] entre todos los objetos del cluster C_i y el centroide c_i, definida como:

$$E = \sum_{i=1}^{k} \sum_{p \in C_i} dist(p, c_i)^2, \tag{3}$$

donde E es la suma del error cuadrático de todos los objetos pertenecientes al cluster $p_i \in C_i$. La finalidad de esta operación es asegurar que los k cluster resultantes sean lo más compactos y distantes posible.

Es aquí, en el cálculo de la varianza donde la optimización de este método se vuelve computacionalmente complicada, pues en el peor de los casos, el total de posibles clasificaciones depende del número total de clusters de manera exponencial y posteriormente habría que comprobar los valores de la varianza dentro de cada cluster, siendo un problema de complexión NP-complejo[15]. Para resolverlo, se utilizan algoritmos voraces[16] como k-Means, que es simple y usado habitualmente.

Este algoritmo sigue una serie de pasos para formar los clusters y evaluar su calidad. Primero, selecciona n objetos dentro del conjunto de datos D para representar los clusters iniciales; y puesto que son los únicos integrantes de ellos, el cálculo del centroide da como resultado el propio objeto. Posteriormente, asigna cada uno de los objetos restantes al cluster con el que es más similar basándose en el cálculo de la distancia Euclidiana entre el objeto y representante del cluster. A partir de aquí, el algoritmo intenta mejorar los resultados de la varianza,

[13] Tipo de coordenadas ortogonales usadas en espacios Euclídeos que tienen como referencia los ejes ortogonales.

[14] Técnica para calcular la diferencia entre el estimador y lo que se estima.

[15] No se definir esto bien

[16] Estrategia de búsqueda en la que se sigue una heurística que consiste en escoger la opción más optima en cada paso de manera local, esperando llegar a una solución general óptima.

para ello, vuelve a calcular el centroide de cada cluster con los nuevos objetos añadidos en la iteración previa, y reagrupa los objetos ahora con los representantes de cada cluster actualizados. El proceso continua hasta que los centroides, y por ende, clusters de la iteración actual sean idénticos a los de la iteración previa.

El pseudocódigo de k-Means queda resumido en el algoritmo 1.

Algoritmo 1: k-Means, basado en la media

Entrada: k: número de clusters,
D: conjunto de datos con n objetos.
Salida: Un conjunto de k clusters.

```
1 inicio
2     selección arbitraria de k objetos de D
        como representantes de los clusters
        iniciales
3     repetir
4         calcular la distancia de cada objeto al
            centroide de cada cluster
5         (re)asignar cada objeto al cluster con
            el que es más similar en función de
            la distancia obtenida
6         actualizar los centroides de los
            clusters con los nuevos objetos
            añadidos
7     hasta no hay cambios
8 fin
```

Sin embargo, este método no garantiza converger hacia un máximo global, sino que muchas veces termina en un máximo local. El resultado depende de la selección de clusters iniciales sobre los que se trabaja, por lo que es es común realizar varias pruebas del algoritmo con diferentes clusters iniciales hasta encontrar un resultado óptimo.

Por otro lado, la complejidad temporal de k-means es considerablemente baja $O(nkt)$, pues depende del número de objetos n, del de clusters k y de la cantidad de iteraciones t y normalmente encontramos que $k \ll n$ y $t \ll n$. Lo cual sugiere que escalar este algoritmo a grandes conjuntos de datos es asequible.

Asimismo, este algoritmo solo puede ser utilizado cuando podemos calcular la media de varios objetos para encontrar el centroide, por lo que en casos donde los datos tengan atributos nominales, no se puede aplicar. Pero existen modificaciones y versiones de k-means que solucionan algunos de estos problemas, como k-modas, que en vez d utilizar la media para calcular el centroide, utiliza la moda, el atributo que más se repite para una característica, y si permite trabajar con datos nominales.

Otra de las desventajas que presenta y que comentábamos al inicio de la sección, es el hecho de que no cumple uno de los requisitos de los métodos de clustering, y es que k-means pide al usuario el número k de clusters a generar como dato de entrada indispensable. Si bien es cierto que se han propuesto estudios para solventar este problema intentando encontrar el mejor valor para k ejecutando varias veces el algoritmo con distintas unidades, pero supone un alto coste computacional.

Además, recordemos que este algoritmo flaquea a la hora de encontrar clusters con formas poco convencionales y de tamaños dispares, siendo realmente sensible a datos anómalos e impurezas, que pueden alterar los cálculos de los centroides.

4.2. ***k-Medoids***

Como se ha mencionado en el apartado previo 4.1, k-means es realmente susceptible a datos anómalos e impurezas, pues como estos objetos se suelen encontrar alejados de todos los datos, cuando se asignan a un cluster y se recalcula el valor del centroide, este valor se ve altamente alterado, lo cual también afecta a la formación del resto de clusters.

Con el objetivo de ser menos sensible con estos datos, k-medoids utiliza uno de los objetos del cluster en vez de su centroide como representante del mismo, y el resto de objetos se añaden al cluster cuyo objeto representativo sea más similar. Por lo que el método para comprobar la calidad del cluster ya no es la varianza, sino el error absoluto, cuya fórmula es la siguiente:

$$E = \sum_{i=1}^{k} \sum_{\in C_i} dist(p, o_i)^2, \tag{4}$$

, siendo E la suma del error absoluto para todos los puntos p_i pertenecientes al cluster C_i ($p_i \in C_i$) con respecto al objeto representativo de dicho cluster o_i. De esta manera, k-medoids agrupa n objetos en k clusters minimizando el error absoluto.

Los algoritmos más conocidos que utilizan esta aproximación para formar clusters son Partioning Around Medoids(PAM) y Clustering LARge Applications (CLARA), cada uno con sus propias características, que veremos a continuación.

4.2.1. ***PAM***

Este algoritmo, que podría traducirse al español como Particionamiento Sobre Medoids, es una de las aplicaciones más extendidas del método k-medoids, y tiene un funcionamiento similar a k-means en el sentido de que utiliza también una heurística voraz de forma iterativa para la formación de clusters y disminuir la complejidad computacional.

Y de la misma forma, el paso inicial también es seleccionar de manera aleatoria k objetos para formar los clusters iniciales, denominados "semillas", y asignar los objetos restantes al cluster cuyo objeto representativo (medoid) es el más similar. Posteriormente, con los nuevos clusters, se comprueba si cambiando el objeto representativo del cluster actual o_i por otro objeto cualquiera dentro del cluster $o_{cualquiera}$ se mejora la calidad del cluster, y en caso de que sea así, se intercambia uno con otro. Esta comprobación se realiza para todos los objetos dentro del cluster que no son el representativo $o_{cualquiera} \neq o_i$ y para todos los clusters, hasta que no se puede mejorar la situación actual. Una vez ha terminado este proceso, se vuelve a calcular la similaridad entre todos los objetos y los representantes de los clusters y se reagrupan si es necesario. El algoritmo termina cuando los clusters resultantes de la iteración actual son idénticos a los de la previa.

Para determinar la calidad del cluster y decidir si un representante de un cluster debe ser cambiado por otro se calcula la distancia entre todos los puntos de un cluster y su objeto representante actual o_i mediante la fórmula 4, cuyo resultado es almacenado. Luego, de forma iterativa cada objeto perteneciente al cluster se convierte en su representante o_j temporal y se vuelve a calcular la distancia entre el resto de puntos y el representante temporal, y su resultado también se almacena, de forma que, una vez todos los puntos han sido representantes, se escoge como representante aquel cuyo resultado ha sido el más bajo, es decir, el más cercano a la mayoría de puntos del cluster, y si este nuevo punto es distinto al representante actual $o_i \neq o_j$, o_j se convierte en el nuevo representante del cluster.

Cuando un intercambio de representante ocurre en un cluster, se debe comprobar si es necesario reestructurar los clusters y el coste que esto conlleva, pues puede haber puntos cuyo cluster más cercano haya variado. Por lo que para cada punto del espacio de objetos se calcula la distancia a los nuevos representantes y se selecciona el que esté más cerca, pudiendo ser un nuevo cluster o el mismo en el que se encontraba pero con el nuevo representante. Una vez seleccionado el cluster más cercano para cada objeto, se calcula el coste con la misma función 4, y si el resultado de esta reestructuración es negativo, es decir, si en la nueva situación los clusters representan mejor a los objetos porque el representante está más cerca de los puntos, se acepta este cambio y se considera que ha aumentado la calidad de los clusters.

Algoritmo 2: PAM, basado en representantes

Entrada: k: número de clusters,
D: conjunto de datos con n objetos.
Salida: Un conjunto de k clusters.

```
1  inicio
2      selección arbitraria de k objetos de D
         como representantes o_i de los clusters
         iniciales
3      repetir
4          calcular la distancia de cada objeto al
             representante de cada cluster
5          (re)asignar cada objeto al cluster con
             el que es más similar en función de
             la distancia obtenida
6          seleccionar un objeto cualquiera del
             cluster que no sea el represente
             actual o_cualquiera
7          calcular el coste T de intercambiar el
             representante del cluster o_i con
             o_cualquiera
8          si T<0 entonces
9              intercambiar o_i con o_cualquiera
                 como representante del cluster
10     hasta no hay cambios
11 fin
```

Este procedimiento, que queda resumido en el algoritmo de PAM 2, es más robusto que k-means al ser más tolerante a datos anómalos e impurezas, porque el medoid está menos influenciado por estos que la media. Sin embargo, esta aproximación es mucho más compleja computacionalmente para cada iteración y sigue solicitando el número de clusters k, por lo que para gŕandes volúmenes de datos resulta ineficiente, dando origen a otra versión enfocada precisamente a este tipo de trabajo: CLARA.

4.2.2. ***CLARA***

CLARA es una variación de PAM y que también aplica k-medoids a la hora de formar clusters, pero que en lugar de tener en cuenta todos los datos n del conjunto de datos D, este algoritmo escoge una muestra aleatoria m de datos y posteriormente aplica PAM para calcular los mejores representantes de la muestra, que idealemente, representa de una manera precisa a D siempre y cuando cada objeto p_i tenga las mismas posibilidades de ser escogido para la muestra. De esta manera y tras varias iteraciones seleccionando muestras arbitrarias, CLARA devuelve la mejor clusterización obtenida de todas ellas.

Esto permite que la complejidad computacional disminuya, pues la comprobación de la calidad de los clusters y el intercambio de representantes que tanto costaba en PAM es reducido de manera drástica con este enfoque, permitiendo de esta manera que pueda trabajar con grándes volúmenes de datos.

Sin embargo, este algoritmo depende totalmente de las muestras escogidas y la probabilidad de que el mejor representante de cada cluster se encuentre en ella, puesto que si tan solo uno de ellos no aparece en alguna de las muestras, CLARA nunca encontrará la mejor clusterización posible.

5. Métodos jerárquicos

Mientras que los métodos basados en particiones cumplen con el requisito básico de agrupar los datos en clusters exclusivos, hay ocasiones en las que clasificar los datos en diferentes niveles, es decir, de forma jerárquica, puede proporcionar grandes ventajas. De esta manera, los métodos jerárquicos organizan los datos en una jerarquía o árbol de clusters.

Útil en escenarios donde los datos tengan de por sí cierta estructura jerárquica (estructura de empleados) o en aquellos en los que se pretenda descrubir si existe una oculta(teroría de la evolución), estos métodos se clasifican a su vez en dos grupos: aglomerativos y divisivos.

- **Métodos jerárquicos aglomerativos**: Estos métodos utilizan una estrategia de "abajo a arriba", partiendo de que cada objeto compone su propio cluster y de forma iterativa los va uniendo en clusters más grandes hasta que solo queda un único cluster, que se conoce como "raiz". En cada iteración se mergen los dos clusters más cercanos entre sí para formar uno solo.
- **Métodos jerárquicos divisivos**: Este tipo de métodos utilizan una estrategia de "arriba a abajo", donde situa a todos los objetos en un cluster inicial, la raiz, y de manera progresiva divide la raiz en subclusters más pequeños sucesivamente hasta que cada cluster queda formado únicamente por un objeto o los objetos de cada cluster son prácticamente idénticos entre sí.

Como se puede ver, son métodos opuestos pero que operan bajo el mismo principio. Sin embargo, es mucho más complicado dividir los datos que juntarlos, pues hay muchas más posibilidades y por lo tanto más comprobaciones que realizar; dando lugar a que los métodos aglomerativos sean más comunes y los divisivos dependan de métodos heurísticos que pueden dar resultados inexáctos.

Los métodos jerarquicos pueden sufrir dificultades a la hora de dividir o juntar clusters, una parte crítica en su procedimiento, puesto que, a diferencia de los métodos basados en particiones donde en cada iteración se reagrupaban los puntos hacia el cluster más cercano, si un método jerárquico divide o junta dos clusters, no hay vuelta atrás y los siguiente pasos trabajarán con el resultado obtenido sin poder cambiarlo. Por lo que decidir cómo y por qué motivo se genera un cluster es una tarea complicada, que de no ser correcta, puede ocasionar resultados pobres. Y la otra desventaja principal de este enfoque es que no escalan bien hacia grandes volúmenes de datos, puesto que deben realizar muchas comprobaciones cuando se queieren unir o separar clusters.

Para lidiar con estos problemas se han desarrollado técnicas como BIRCH y Chameleon de las que se habla en las secciomes 5.4 y 5.5, que incorporan métodos jerárquicos con otros métodos, resultando en otros tipos de clustering conocidos como multi fase y microclustering.

Antes de explicar algunos de los algoritmos más utilizados dentro de este grupo, AGNES 5.2, DIANA 5.3, BIRCH 5.4 y Chameleon 5.5, analizaremosmos brevemente las diferentes maneras que existen en los métodos jerárquicos para medir distancias.

5.1. ***Medidas de distancia***

Independiéntemente del tipo de método que se utilice, aglomerativo o divisivo, se necesita una

forma de medir la distancia entre dos clusters.

Hay principalmente tres formas diferentes de realizar este cálculo, aunque existen otras variaciones en las que, por ejemplo, se consideran los centroides, etc.

- **Distancia mínima**: Esta define la proximidad entre dos clusters como la distancia entre los puntos más cercanos entre ambos, también conocida como "single link".

$$dist_{min}(C_i, C_j) = \min_{p \in C_i, p' \in C_j} \{|p - p'|\} \quad (5)$$

- **Distancia maxima**: Define la proximidad entre dos clusters como la distancia entre los puntos más alejados entre ellos, también conodida como "complete link".

$$dist_{max}(C_i, C_j) = \max_{p \in C_i, p' \in C_j} \{|p - p'|\} \quad (6)$$

- **Distancia media**: Define la proximidad entre dos clusters como la distancia media entre todos los puntos de ambos clusters.

$$dist_{avg}(C_i, C_j) = \frac{1}{n_i \cdot n_j} \sum_{p \in C_i, p' \in C_j} |p - p'| \quad (7)$$

Las dos primeras son medidas que se van a los extremos y son susceptibles a datos anómalos e impurezas, por lo que el uso de la media es un compromiso entre ambas para solventar este problema.

5.2. *AGNES*

AGglomerative NESting (AGNES) es quizás el algoritmo jerárquico aglomerativo más básico y extendido, pues aplica de forma exácta el concepto mencionado previamente. Parte de una separación de todos los objetos, cada uno formando un único cluster y compara la proximidad entre todos ellos utilizando el método que más convenga de los mencionados en la sección previa 5.1. Selecciona aquellos dos con el resultado más pequeño, es decir, los dos más cercanos y los fusiona en un solo cluster, y se vuelven a calcular la similaridad entre todos los clusters, así sucesivamente hasta que todos los datos quedan agrupados en la raiz [18].

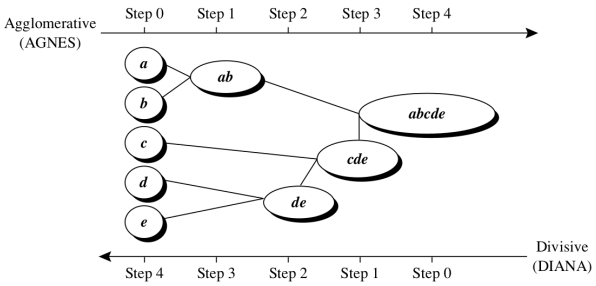

Figure 1: Clusterización jerárquica algomerativa (AGNES) y divisiva (DIANA) sobre el conjunto de datos {*a*,*b*,*c*,*d*,*e*} [16].

5.3. *DIANA*

El algoritmo DIvisive ANAlysis (DIANA) es la contraparte a AGNES, siendo también el método jerárquico divisivo más simple y utilizado. En este caso, al ser divisivo, parte de un solo cluster en el que se encuentran todos los datos y que de forma iterativa los va dividiendo en función de un criterio preestablecido, como por ejemplo la distancia Euclididana mínima entre los objetos más cercanos del vencidario.

Tanto AGNES como DIANA generan una estructura denominada dendograma que se puede ver en la figura 5.2, donde se compara el proceso de estos dos algoritmos.

5.4. *BIRCH*

[16, 19, 20] El algoritmo BIRCH, proveniente del nombre Balanced Iterative Reducing and Clustering using Hierarchies, es uno de los que mencionábamos previamente que combina el método jerárquico con otros métodos para solucionar los problemas de escalabilidad e imposibilidad de deshacer las divisiones o fusiones de clusters.

Lo que permite a BIRCH esto es utilizar un concepto denominado *característica de agrupamiento*, en inglés clustering feature (CF), que sirve para resumir un cluster, y otro llamado *árbol de características de agrupamiento*, en inglés clustering feature tree (CF-tree), para representar la jerarquía de clusters. Estos dos atributos ayudan en la escalabilidad y rapidez del algoritmo y hacen posible utilizarlo en conjuntos de datos en los que se añaden nuevos datos. Veamos brevemente cómo funciona.

Consideremos un cluster C_i con n objetos en él, CF es un vector de tres dimensiones que pretende

resumir la información estadística del cluster de la siguiente manera

$$CF = (n, LS, SS) \tag{8}$$

, donde LS es la suma de los n objetos y SS es la suma de los n objetos elevados al cuadrado. Esto permite obtener datos como el centroide c_i, el radio R y diámetro D del mismo, donde estos dos últimos representan lo ajustado que está el cluster alrededor del centroide.

Usando esto, no hace falta guardar todos los datos del cluster, sino únicamente esta variable. Y a la hora de fusionar dos clusters C_i y C_j, sus CF respectivos se suman tal que

$$CF_i + CF_j = (n_i + n_j, LS_i + LS_j, SS_i + SS_j) \tag{9}$$

.

Por ejemplo, suponiendo que tenemos un cluster C_i con los puntos (2,5), (3,2) y (4,3), CF_i se calcula de la siguiente forma:

$$\begin{aligned} n_i &= (3)\,, \\ LS_i &= (2+3+4, 5+2+3) = (9, 10)\,, \\ SS_i &= \left(2^2+3^2+4^2, 5^2+2^2+3^2\right) = (29, 38)\,, \\ CF_i &= (n_i, LS_i, SS_i) = (3, (9, 10)\,, (29, 38))\,. \end{aligned}$$

Y ahora, consideremos que hay que fusionar C_i con C_j para formar otro cluster nuevo C_z, siendo $CF_j = (3, (6, 7)\,, (36, 49))$,

$$\begin{aligned} &CF_z = CF_i + CF_j = \\ &= (3, (9, 10)\,, (29, 38)) + (3, (6, 7)\,, (36, 49)) = \\ &= (6, (15, 17)\,, (65, 87))\,. \end{aligned}$$

Y el otro concepto, CF-tree tiene una estructura en forma de árbol que contiene el CF de cada cluster. Una estructura de árbol está formada por raíces y hojas. Un nodo raíz debe tener al menos un descendiente u hoja, mientras que las hojas son nodos terminales. Las hojas almacenan el CF de un cluster, y su nodo padre o la raíz de dicha hoja almacena la suma de los CF de todos sus hijos. Este utiliza dos parámetros para controlar el desarrollo del árbol: el factor de ramificación B, y el umbral T; estos indican cuántos hijos u hojas puede tener un nodo y cuántos nodos puede tener un cluster, respectivamente. El tamaño del umbral determina el tamaño del árbol y su elección supone cierta dificultad.

BIRCH es un método que consiste de dos fases principales:

1. BIRCH escanea todo el conjunto de datos con la intención de formar un CF-tree inicial para comprimir todos los datos intentando mantener la estructura inherente de los datos. Esta fase es dinámica y el CF-tree es generado conforme se van insertando más datos, que se van insertando en las hojas siempre y cuando estas no superen ya el umbral T de objetos permitidos, en cuyo caso, el cluster se divide en dos. Si llegase un punto en que tantos nodos no cupiesen en la memoria disponible, se aumenta el valor de T, permitiendo que los nodos alberguen más objetos y haya menos información que almacenar, provocando una reconstrucción de CF-tree.
2. Posteriormente, se aplica un algoritmo de clusterización a las hojas del CF-tree que elimina clusters con poca calidad o pocos datos, que suelen ser datos anómalos y fusiona grupos de datos densos en clusters aún más grandes. Típicamente son algoritmos basados en particiones.

Esta aproximación de clusterización ha probado ser rápida, escalable y fiable con los resultados generados, sin embargo sigue teniendo problemas, puesto que el umbral T determina en gran medida la formación de clusters y puede que se generen grupos que no son precisamente "naturales" y por otro lado, como utiliza el radio como criterio para definir clusters tiene peor resultados si los clusters no son esféricos o circulares.

5.5. ***Chameleon***

Chameleon es un algoritmo jerárquico que utiliza un modelado dinámico para calcular la similaridad entre pares de clusters. Esto es determinado por cómo de bien conectados estén los objetos dentro del cluster y la proximidad entre clusters; es decir, dos clusters se fusionarán si su interconectividad es alta y están muy cerca. Una de las principales ventajas de esta aproximación es que puede adaptarse a los clusters y sus diferentes formas para cualquier tipo de dato, siempre que se pueda aplicar una función para calcular la similaridad.

En la imagen 5.5 podemos observar el funcionamiento de este algoritmo. Primero utiliza k-vecinos para construir un grafo disperso, donde cada vértice representa un objeto y estos están conectados mediante aristas si se encuentran dentro del rango de k-vecinos y son similares. Las aristas,

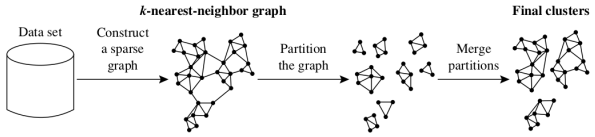

Figure 2: Chameleon: clusterización jerárquica basada en k-vecinos y modelado dinámico [16].

además, están ponderadas en función de la similaridad de los objetos.

Posteriormente, se aplica un algoritmo basado en particiones para dividir el grafo generado por k-vecinos en clusters, eliminando aquellas aristas con menos peso, es decir, las que unen clusters menos similares. Y una ver termina este proceso, se vuelve a aplica un algoritmo jerárquico aglomerativo que itera sobre los clusters generados y los fusiona basándose en la *interconectividad relativa* y la *cercanía relativa* de los clusters.

Este algoritmo ha demostrado se capaz de identificar clusters con formas arbitrarias de mejor manera que otros algoritmos, pero su coste computacional sigue siendo elevado cuando se trabaja con conjuntos de datos con muchos atributos.

5.6. ***Métodos jerárquicos probabilísticos***

Todos los métodos jerárquicos vistos hasta ahora utilizan algoritmos y medidas de proximidad para clusterizar, lo cual hace que sean fáciles de entender y eficientes en muchos ámbitos. Sin embargo, tienen varios problemas con los que todavía se intenta lidiar:

- Establecer una distancia de proximidad para los métodos jerárquicos es complicado en la gran mayoría de los casos.
- Para aplicar un algoritmo, es necesario que los objetos tengan todos los datos si se desea obtener un resultado de alta calidad.
- Muchos de estos métodos utilizan heurísticas que buscan optimizaciones locales fusionando o separando clusters, pudiendo resultar en un resultado gobal inexacto.

Para ello, los métodos jerárquicos probabilísticos pretenden solventar algunos de estos problemas utilizando modelos probabilísticos para calcular la proximidad entre clusters.

Estos métodos tratan de ver el conjunto de datos como una muestra de un mecanismo subyacente que ha generado los datos, denominado *moldeo generativo*, y lo que pretende esta forma de clustering es intentar estimar de la manera más precisa el modelo generativo mediante los datos que se van a procesar.

Normalmente se asume que estos modelos generativos siguen funciones de distribución como la de Bernoulli o la Gausiana, por lo que se reduce mucho el campo de trabajo y solo queda probar valores hasta dar con los que más se ajustan a los objetos del conjunto de datos.

6. Métodos basados en densidad

Los métodos jerárquicos y basados en particiones son muy útiles y eficaces encontrando clusters con formas esféricas y circulares, y aunque algunos de ellos sí puedan detectar en cierta medidad otras formas, sigue siendo insuficiente, y en conjuntos de datos con datos anómalos e impurezas es muy probable que no sean capaces de generar clusters de calidad.

Una solución a este problema es modelar los clusters como regiones densas de objetos separadas por regiones de tenue densidad, la estrategia principal de los algoritmos basados en densidad.

A continuación explraremos los tres algoritmos más populares dentro de este grupo, DBSCAN 6.1, OPTICS 6.2 y DENCLUE 6.3.

6.1. ***DBSCAN***

El algoritmo DBSCAN (Density-Based Spatial Clustering of Applications with Noise), propuesto por Martin Ester, Hans-Peter Kriegel, Jorg Sander y Xiaowei Xu en 1996 [21], se basa fundamentalmente en encontrar objetos en cuyos alrededores haya más objetos y conectarlos con objetos de iguales características para formar clusters, dejándo como separación entre ellos áreas con pocos datos.

Este algoritmo solicita dos valores al usuario, el primero $\epsilon > 0$ delimita el radio del vecindario de todos los objetos, y el segundo *MinPts* es utilizado para deteminar la densidad del vecindario de un objeto o. Cuando el vecindario de un objeto o, definido como $N_\epsilon(o) = \{o \in D | dist(o, q) \leq \epsilon\}$, contiene más de *MinPts* objetos, o se considera un objeto central o núcleo, que es el centro o pilar de un cluster.

Sin embargo, si solo utilizáramos esto para formar los clusters, tendríamos problemas con interferencias y datos anómalos, pues los puntos en los

bordes tienen menos objetos a su alrededor y se podría dar una forma de cluster errónea. Para evitar eso, se requiere que, para cada punto p de un cluster C, haya otro punto q en C tal que p esté dentro del vecindario de q ($p \in N_\epsilon(q)$) y que en el vecindario de q haya al menos $MinPts$ objetos ($N_\epsilon(q) > MinPts$). Si se cumple esta condición, se considera que p es directamente alcanzable por densidad desde q.

Esta condición suele ser recíproca entre núcleos, pero no entre núcleos y puntos fronterizos, tal y como se aprecia en la figura 6.1

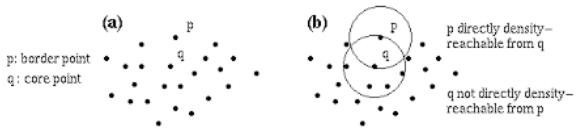

Figure 3: DBSCAN: diferencia entre núcleos y puntos fronterizos [21].

De esta definición se puede extrapolar otra, y es que un punto p es alcanzable por densidad desde q, si hay una cadena de puntos $p_1, \cdots, p_n$ donde el primer punto es q ($p_1 = q$) y $p_i + 1$ es directamente alcanzable por densidad desde p_i. Al igual que la anterior, no suele ser recíproca con núcleos y puntos fronterizos, pero sí entre núcleos. Sin embargo, puede que existan dos puntos fronterizos del mismo cluster que no sean alcanzables por densidad entre ellos porque no tengan $MinPts$ objetos en su vecindario, pero que si lo sean desde un núcleo del cluster, que da lugar a otra definición.

Se dice que un punto p está conectado por densidad a un punto q si hay otro punto o desde el cuál, ambos, p y q, son alcanzables por densidad, y esta propiedad si es recíproca siempre.

Ejemplos de estas tres definiciones pueden observarse en la figura 6.1.

Con estos conceptos, DBSCAN es capaz de formar clusters de forma que todos sus objetos estén conectador por densidad y detectando como impurezas y datos anómalos aquellos puntos que no pertenezcan a ningún cluster.

Asi pues, el primer paso de este algoritmo es seleccionar un punto cualquiera p del conjunto de datos D, y comprueba todos los puntos alcanzables por densidad con respecto a ϵ y $MinPts$. Si p es un núcleo, se forma un cluster con todos los objetos alcanzables por densidad desde p que no pertenezcan a otro cluster, y si es un punto fronterizo, no hay

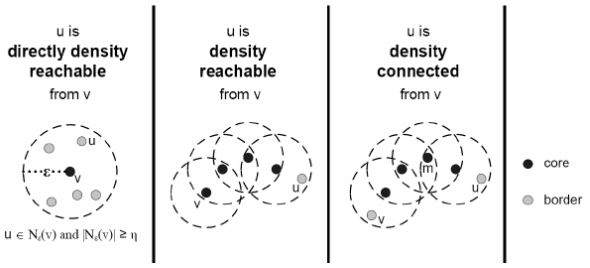

Figure 4: DBSCAN: términos *directamente alcanzable por densidad*, *alcanzable por densidad* y *conectado por densidad* [22].

puntos que sean alcanzables por densidad desde p, por lo que es marcado como ruido y selecciona otro objeto.

Una vez ser forma el cluster, se selecciona otro objeto que no se haya visitado y se repite la operación. Asimismo, DBSCAN puede fusionar dos cluster si se encuentran muy cerca entre sí.

Este procedimiento queda reflejado en el algoritmo 3.

6.2. ***OPTICS***

Sin embargo, DBSACN sigue solicitando al usuario ϵ y $MinPts$, valores cruciales en la formación de clusters y de difícil elección. Como solución a este problema y sin la necesidad de introducir parámetros de entrada, surgió OPTICS (Ordering Points to Identify the Clustering Structure), que no devuelve una clusterización, sino un orden de clusters (cluster ordering); una lista linea de todos los objetos del conjunto de datos y que representa la estructura de clusterización basada en la densidad de los datos. Aquellos objetos que sean próximos entre sí estarán más cerca en la lista de ordenación.

Con este proceso podemos extraer la información básica de la clasificación, como los centros de los clusters o grupos con formas raras, una visualización de la clusterización y también se puede derivar la estructura intrínseca de la agrupación.

Para ello, OPTICS necesita obtener dos datos adicionales para cada objeto:

- **Distancia al núcleo**: La distancia al núcleo de un objeto p es el valor más pequeño para ϵ' de manera que en el vecindario delimitado por dicho valor haya al menos $MinPts$, y si p cumple con esta condición, se le considera

Algoritmo 3: DBSCAN, basado en densidad

Entrada: D: conjunto de datos con n objetos,
ϵ: radio del vecindario,
$MinPts$: umbral de densidad del vecindario.
Salida: Un conjunto de clusters basados en densidad.

```
1  inicio
2      marcar todos los objetos como no visitados
3      repetir
4          seleccionar un objeto p no visitado
5          marcar p como visitado
6          si el ε vecindario de p tiene al menos MinPts objetos entonces
7              creamos un nuevo cluster C y le añadimos p
8              sea N el conjunto de objetos en el ε vecindario de p
9              para cada punto p' ∈ N hacer
10                 si p' no ha sido visitado entonces
11                     marcamos p' como visitado
12                     si el vecindario de p' tiene al menos MinPts entonces
13                         añadimos los puntos del vecindario de p' a N
14                     si p' no pertenece a ningún cluster entonces
15                         añadimos p' a C
16             fin
17             devolvemos C
18         en otro caso
19             marcamos p como ruido
20     hasta quden objetos por visitar
21 fin
```

núcleo. En caso de que que haya $MinPts$ en el ϵ' vecindario, se considera que la distancia al núcleo es indefinida.

- **Distancia de alcance**: La distancia de alcance hacia un objeto p desde otro objeto q es el radio mínimo que hace que p sea alcanzable por densidad desde q (6.1), por lo que q debe ser un núcleo y p debe estar en su vecindario. Si q no es un núcleo, se considera que la distancia de alcance de p a q es indefinida.

  Es posible que un objeto p tenga más de una distancia de alcance, es decir, que sea alcanzable desde varios puntos. En este caso, la medida que más nos interesa es la distancia de alcance mínima.

  Se puede formalizar este concepto como $\max\{distancia\ al\ nucleo(q), dist(p,q)\}$.

OPTICS almacena para cada objeto del conjunto de datos la distancia al núcleo y una distancia de alcance apta, y en función de la distancia de alcance de cada objeto desde su núcleo más cercano, genera una lista llamada OrdenSemillas (OrderSeeds).

OPTICS escoge un objeto cualquiera como punto actual p, obtiene su ϵ vecindario, calcula su distancia al núcleo y a priori establece su distancia de alcance como indefinida. Si p no es un núcleo, simplemente pasa al siguiente objeto de OrdenSemillas. Si p es un núcleo, actualiza la distancia de alcanza para cada objeto q de su ϵ vecindario y lo inserta en OrdenSemillas si todavía no está en la lista. Esta iteración continúa hasta que la OrdenSemillas está vacía.

6.3. ***DENCLUE***

Los dos algoritmos previos, DBSACN y OPTICS (6.1, 6.2) presentan un inconveniente, ϵ. Pequeñas variaciones en el radio del vecindario de los objetos puede resultar en clusters totalmente distintos. DENCLUE [23] (DENsity based CLUstEring) pretende solventar este problema utilizando un enfoque de estimación de densidad no paramétrica llamado estimación de densidad nuclear.

La idea principal detrás de este algoritmo está basada en la idea de que la influencia de cada punto, es decir, el impacto que este tiene en su vecindario, puede ser modelada formalmente utilizando una función matemática denominada *función de influencia*, con la que podemos obtener la densidad media del espacio de datos, sumando la función de influencia de todos los puntos.

De esta manera, los clusters pueden ser definidos encontrando los denominados *atractores de densidad*, puntos máximos locales de la función de densidad media. Estos se pueden hallar mediante el algoritmo de Escalada Simple[17] guiado por el gradiente de la propia función. El uso de la función de

[17] Técnica de optimización matemática utilizada para encontrar óptimos locales.

densidad media también facilita encontrar clusters de formas arbitrarias.

DENCLUE implementa de manera eficaz este planteamiento y lo hace más eficaz, pues no todos los puntos del espacio de datos contribuyen de forma notable en la función de densidad media, por lo que en su lugar, usa una función local de densidad, teniendo en cuenta solo los puntos que influyen en la función. En el primer paso de este algoritmo, se preprocesan los datos, generando un mapa de los objetos relevantes del espacio de datos que sirve para aumentar la velocidad de cálculo de la función de densidad. El segundo paso es el propio clustering, donde el identifica los atractores de densidad y aquellos puntos a los que influye.

Este algoritmo tiene varias ventajas, y es que es tan versátil como otros algoritmos como DBSACN y k-means, pero además tolera extremadamente bien datos anómalos e impurezas.

7. Métodos basados en rejilla

Todos los métodos expuestos hasta ahora tienen un enfoque sobre los datos, es decir, se basan en la distribución de los mismos en el espacio de datos para formar clusters. Sin embargo, los métodos basados en rejilla se basan en el espacio; no dividen los datos, sino que distribuyen el espacio de datos en celdas, independientemente de cómo estén repartidos los datos.

Estos métodos dividen el espacio de datos en un número finito de celdas que forman una estructura de rejilla o cuadrícula, donde se realizan las operaciones de clusterización, aumentando considerablemente la velocidad de clustering, hasta el punto que la cantidad de datos es irrelevante, solo depende del número de celdas.

En esta sección veremos dos de los métodos más conocidos dentro de este grupo, STING 7.1 y CLIQUE 7.2.

7.1. ***STING***

La técnica STING (STatistical INformation Grid) es una técnica de clusterización en la que el espacio de datos es dividido en celdas rectangulares, pudiendo descomponerlo de forma jerárquica. Cada capa de celdas forma un nivel distinto, done cada celda de un nivel superior está compuesta por varias celdas del nivel inferior, lo cual permite obtener esa estructura jerárquica de los datos.

De cada celda se calcula de antemano información estadística clave de los datos que esta contiene, tales como la media, máximo, mínimo; esto facilita la obtención de paramentos para celdas superiores: el número de objetos dentro de la celda, la media, desviación típica[18], máximo, mínimo y si los datos siguen algún tipo de distribución (normal, uniforme, exponencial o ninguna).

STING selecciona una de las capas superiores que se han generado, no necesariamente la primera, sino una con pocas celdas. Examina todas ellas y descarta aquella celdas que tras analizarlas, se considere que tienen poca relevancia con respecto a la clasificación deseada. Posteriormente, desciende un nivel y examina las celdas inferiores de las celdas que han superado la prueba y realiza el mismo proceso de manera iterativa hasta que se alcanza la última capa. Si en esta capa se considera que se cumple con los requisitos de la clusterización, se devuelven las celdas relevantes; por el contrario, se realiza un análisis más en profundidad de las mismas. Asimismo, todas las celdas descartadas son analizadas posteriormente.

Uno de los principales problemas de STING es que dependiendo del tamaño de las celdas, el resultado puede ser más o menos preciso, pues devuelve el polígono resultante de las celdas finales. Cuanto más pequeñas sean las celdas, menos se notarán las uniones bruscas y rectas entre celdas, y si el tamaño de las mismas fuera de 0, el resultado sería idéntico a DBSCAN 6.1, por lo que a veces se considera STING como un algoritmo basado en densidad.

7.2. ***CLIQUE***

En gran cantidad de ocasiones nos encontramos con situaciones en las que algunos de los atributos de los datos no aportan información a la hora de clasificar los datos. Por ejemplo, si tratásemos de clusterizar un conjunto de pacientes de gripe, atributos como “trabajo”, “género” o “edad” varían drásticamente entre los pacientes, lo que genera dificultades a la hora de clusterizar todo el conjunto de datos. En estos casos es más conveniente realizar búsquedas en subespacios, donde se seleccionan los datos relevantes para la clasificación. nE el caso de la gripe, podría ser pacientes con síntomas similares en un rango de edad entre los tres y diez años.

[18] Medida estadística que sirve para cuatificar la dispersión de un conjunto de datos numéricos.

CLIQUE (CLustering In QUEst) es un método simple de clusterización basada en rejilla para encontrar clusters basados en densidad dentro de subespacios. Esta técnica particiona cada atributo de los datos en una dimensión diferente que posteriormente divide en celdas y cataloga como densas o no si la cantidad de objetos dentro de una celda supera cierto umbral.

El primer paso de este algoritmo consiste precisamente en esto, dividir los datos en diferentes dimensiones, generar las rejillas y encontrar las celdas que superen el umbral. Para encontrar estas celdas, se utilizan los ejes. Se ponen ciertos intervalos a los valores del atributo, y si un intervalo supera el umbral, se seleccionan las celdas de ese intervalo en las que haya objetos. Si hay varios intervalos contiguos que superan el umbral, estos se unen. Este proceso se repite para cada dimensión. Y en el segundo paso, CLIQUE utiliza las celdas densas para formar clusters que pueden tomar formas arbitrarias.

Algunas de las principales ventajas que presenta es que no es sensible al orden de entrada de los datos, escala linealmente con el tamaño del conjunto de datos y automáticamente encuentra aquellos subespacios con tantos atributos (dimensiones) posibles en los que hay clusters de alta densidad.

8. Evaluación del clustering

Las técnicas y algoritmos que se han expuesto son tan solo los más populares, una pequeña parte de todos los que existen, pues hay múltiples variaciones, nuevas versiones y combinaciones de cada uno de ellos enfocados en mejorar el rendimiento, cubrir flaquezas y permitir nuevas aplicaciones y funcionalidades.

Sin embargo, ¿cómo sabemos que el resultado de una clusterización es preciso y correcto? Es necesario evaluar la clasificación final de un proceso de clustering para cerciorarnos de que el método utilizado y los valores asignados a las variables pertinentes son los adecuados. Gracias a esto, se pueden determinar los puntos fuertes de los algoritmos e identificar aquellas condiciones, valores y tipos de datos bajo los que ofrece mejores resultados.

Para valorar el resultado de una clusterización se suelen realizar tres tareas:

- **Evaluar la tendencia de la clasificación**: Intentar determinar si existe una estructura no aleatoria en la distribución de los datos, pues aplicar un método de clustering a un conjunto de datos aleatorio no devolverá resultados útiles y con significado, pues los clusters son también aleatorios y no aportan información. Para esto, se suele usar un procedimiento similar al de los métodos jerárquicos probabilísticos 5.6 en el que se calcula la probabilidad de que el conjunto de datos haya sido generado por una distribución de datos uniforme. Uno de los procedimientos más comúnes es la Estadística de Hopkins.

- **Determinar el número de clusters**: En algunos algoritmos como k-means 4.1 el número final de clusters es solicitado al usuario y por lo tanto debe aproximarse de antemano. Sin embargo, es una buena práctica calcular este valor antes de utilizar cualquier método para contrastar el resultado final. Esta tarea aunque ciertamente complicada, pues los factores de los que depende son muchos (distribución de los datos, número de atributos, cantidad de datos...) y para la que existen numerosas formas de calcularla, puede ser simplificada para un mayor entendimiento o fácil aplicación. Uno de los métodos más sencillos es aproximar el número de clusters a $\sqrt{\frac{n}{2}}$, donde en cada cluster debería haber $\sqrt{2 \cdot n}$, para un conjunto de n datos. Otros métodos son Elbow Method y Validación cruzada.

- **Calcular la calidad de la clusterización**: Una vez los datos han sido clasificados, se debe verificar que el resultado es bueno. Para ello se pueden usar principalmente dos aproximaciones: comprobar si los clusters acogen bien los datos (métodos intrínsecos) o si los clusters se aproximan a una realidad existente (métodos extrínsecos), que suele ser una clasificación ideal realizada por expertos en la materia.

  Los métodos extrínsecos intentan puntuar la clusterización obtenida C con una realidad C_g basandose en cuatro conceptos fundamentales: homogeneidad del cluster, la integridad del cluster, la pureza del cluster y la preservación de clusters pequeños. Mientras que los métodos intrínsecos se centran en lo compactos que son los clusters y la separación entre ellos calculando distancias entre objetos.

9. Anexo

9.1. Índice

9.2. Algoritmos

9.3. Tablas

9.4. Imágenes y figuras

10. Referencias

[1] Alberts, D. S., & Papp, D. S. (1997). The information age: An anthology on its impact and consequences. Office of the Assistant Secretary of Defense Washington DC Command and Control Research Program (CCRP).
[2] Becoming A Data-Driven CEO — Domo. (2018). Data never sleeps 6.0 https://www.domo.com/solution/data-never-sleeps-6
[3] Xu, Z., & Shi, Y. (2015). Exploring big data analysis: fundamental scientific problems´. Annals of Data Science, 2(4), 363-372.
[4] Definición Data Science apuntes FCD
[5] Definición Data Mining libro 100
[6] The R Project for Statistical Computing. (n.d.). Retrieved from https://www.r-project.org/
[7] Moreno, A. (1994). Aprendizaje automático. Llibre, Edicions UPC.
[8] Apuntes JJ clustering
[9] Alkhaibari, A. A., & Chung, P. (2017). Cluster analysis for reducing city crime rates. 2017 IEEE Long Island Systems, Applications and Technology Conference (LISAT). doi:10.1109/lisat.2017.8001983

[10] Song, Y., Meng, H., & Zhang, Y. (2010). Clustering analysis and its applications. 2010 Second IITA International Conference on Geoscience and Remote Sensing. doi:10.1109/iita-grs.2010.5602787

[11] Prabhu, J., Sudharshan, M., Saravanan, M., & Prasad, G. (2010). Augmenting Rapid Clustering Method for Social Network Analysis. 2010 International Conference on Advances in Social Networks Analysis and Mining. doi:10.1109/asonam.2010.55

[12] Baron, J. N., Aznar, M. N., Monterubbianesi, M., & Martínez-López, B. (2020). Application of network analysis and cluster analysis for better prevention and control of swine diseases in Argentina. PLoS ONE, 15(6), 1–26. https://doi.org/10.1371/journal.pone.0234489

[13] Li, J., & Chen, P. (2008). The application of Cluster analysis in Library system. 2008 IEEE International Symposium on Knowledge Acquisition and Modeling Workshop. doi:10.1109/kamw.2008.4810639

[14] Emebo, O., Aromolaran, O., Oyelade, J., Isewon, I., Oladipupo, O., Omogbadegun, Z., ... Olawole, O. (2019). Data Clustering: Algorithms and Its Applications. 2019 19th International Conference on Computational Science and Its Applications (ICCSA). doi:10.1109/iccsa.2019.000-1

[15] Carugo, O., & Eisenhaber, F. (2010). Data Mining Techniques for the Life Sciences. Humana Press.

[16] Han, J., Kamber, M., & Pei, J. (2012). Data Mining: Concepts and Techniques (3rd ed., p. 740). 225 Wyman Street, Waltham, MA 02451, USA: Morgan Kaufmann Publishers, Elsevier.

[17] Höppner, F., Klawonn, F., Kruse, R., & Runkler, T. (1999). Fuzzy cluster analysis: methods for classification, data analysis and image recognition. John Wiley & Sons.

[18] Sano, A. V., Imanuel, T. D., Calista, M. I., Nindito, H., & Condrobimo, A. R. (2018). The Application of AGNES Algorithm to Optimize Knowledge Base for Tourism Chatbot. 2018 International Conference on Information Management and Technology (ICIMTech). doi:10.1109/icimtech.2018.8528174

[19] T. Zhang. (n.d.). BIRCH: An efficient data clustering method for very large databases. Proc. ACM SIGMOD Conf. Management of Data, 103-114.

[20] Lorbeer, B., Kosareva, A., Deva, B., Softić, D., Ruppel, P., & Küpper, A. (2018). Variations on the Clustering Algorithm BIRCH. Big Data Research, 11, 44-53. doi: 10.1016/j.bdr.2017.09.002

[21] Ester, M., Kriegel, H. P., Sander, J., & Xu, X. (1996, August). A density-based algorithm for discovering clusters in large spatial databases with noise. In Kdd (Vol. 96, No. 34, pp. 226-231).

[22] Schlitter, N., Falkowski, T., & Laessig, J. (2011). DenGraph-HO: Density-based Hierarchical Community Detection for Explorative Visual Network Analysis. Research and Development in Intelligent Systems XXVIII, 283-296. doi:10.1007978-1-4471-2318-7_22

[23] Hinneburg, A., & Keim, D. A. (1998). An efficient approach to clustering in large multimedia databases with noise.