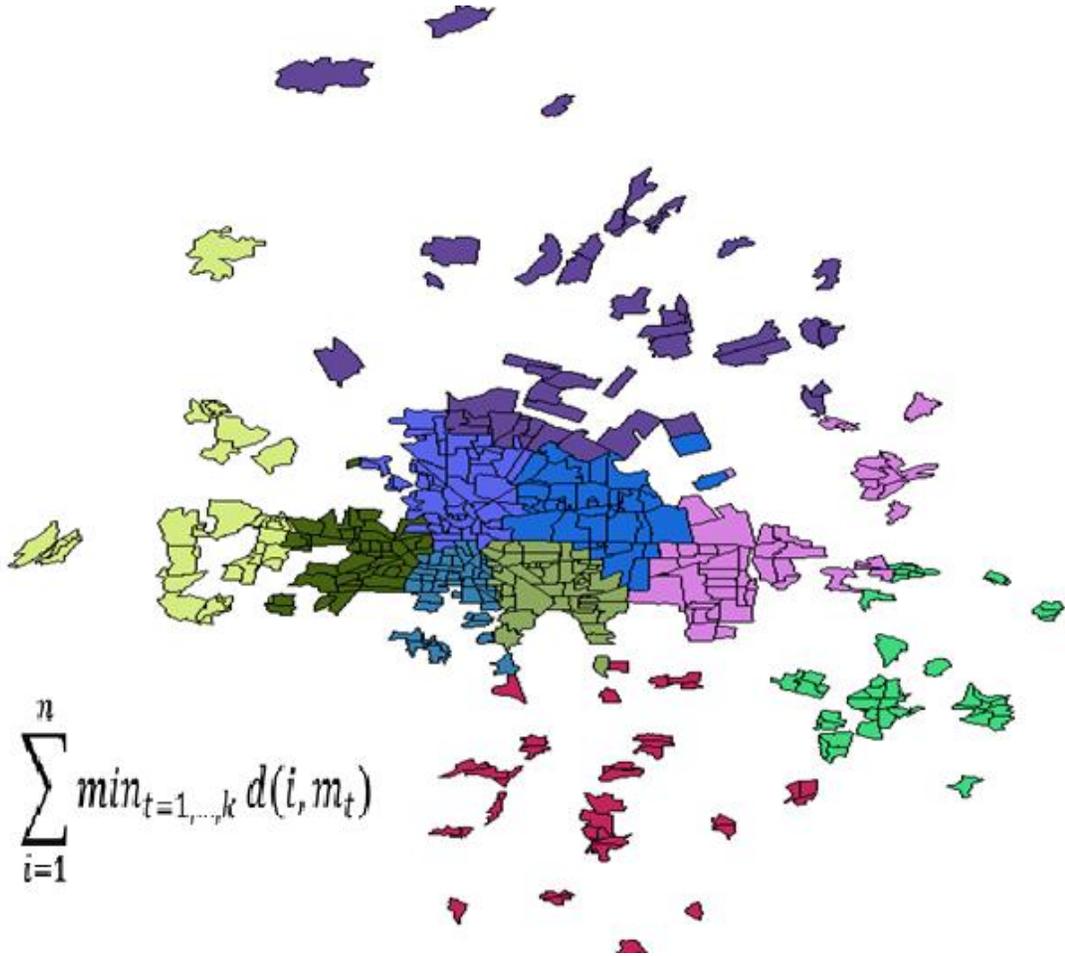


Introducción al Particionamiento



$$\sum_{i=1}^n \min_{t=1, \dots, k} d(i, m_t)$$

Introducción al Particionamiento

Introducción al Particionamiento



Introducción al Particionamiento es una publicación editada por la Universidad Tecnocientífica del Pacífico, S.C. Calle Morelos, 377, Col. Centro, C.P. 63000. Tepic, Nayarit, México. Tel (311)373-9787.

<http://www.tecnocientifica.com.mx>,

http://tecnocientifica.com.mx/editorial_tecnocie/index.php/editorialutp/index

Registro RENIECYT: 1701267

Derechos Reservados © Junio 2022. Primera Edición digital

ISBN

978-607-8759-31-6

Queda prohibida la reproducción total o parcial del contenido de la publicación sin previa autorización de la Universidad Tecnocientífica del Pacífico.

Introducción al Particionamiento

Autor

María Beatriz Bernábe Loranca

Editor

Carmen Cerón Garnica
Rogelio González Velázquez

Prólogo

Rogelio González Velázquez

Comité Evaluador

Esta obra fue evaluada por pares

Dr. Rogelio González Velázquez
Dr. Martín Estrada Analco

Diseño de portada

María Beatriz Bernábe Loranca

Índice

Prólogo	VII
Introducción	IX
Capítulo 1. Preliminares	12
1.1 Conjuntos	12
1.1.1 Conjuntos finitos e infinitos	15
1.2 Clase	15
1.3 Orden en matemáticas discretas.....	16
1.3.1 Conjuntos parcialmente ordenados.....	17
1.4 El orden en los algoritmos.....	18
1.4.1 Algoritmo de ordenamiento	19
1.5 El asunto discreto	22
1.5.1 Relaciones de equivalencia.....	22
1.5.2 Clases de equivalencia	23
1.6 Partición discreta	25
1.7 El quehacer matemático y computacional de crear grupos a partir de un conjunto	26
Capítulo 2. Introducción al Particionamiento	29
2.1 Agrupamiento	29
2.1.1 Algoritmo de agrupamiento	30
2.2 Clúster	34
2.3 Proceso para el Agrupamiento.....	36
2.4 Taxonomía del Agrupamiento	38
2.4.1 Métodos Jerárquicos.....	40
2.4.2 Métodos no jerárquicos.....	42
2.4.3 Métodos no jerárquicos por umbral.....	44
2.5 Enfoque de Particionamiento por división para la optimización	45

2.5.1 Método de <i>k</i> -medias y Forgy.....	47
2.5.2 Método de transferencias.....	48
2.5.3 Método de nubes dinámicas	48
2.6 Suma mínima de distancias al cuadrado en el Particionamiento.....	51
2.6.1 Agrupación mínima de suma de cuadrados: <i>Formulación compacta</i>	52
2.7 Agrupamiento generalizado de suma mínima de cuadrados (MSSC)	54
Capítulo 3. Propuestas tradicionales del Particionamiento.....	57
3.1 Algoritmo K-means (medias).....	57
3.1.1 Procedimiento del Algoritmo	57
3.1.2 Minimización de la suma de errores cuadráticos.....	59
3.1.2.1 Algoritmo ejemplo Algoritmo de Agrupamiento K-means.....	60
3.2 Agrupamiento K-medoids (medoides)	61
3.2.1 Procedimiento del Algoritmo	62
3.2.2 Pseudocódigo general de K-medoides.....	63
3.3 El agrupamiento desde el enfoque supervisado y no supervisado	64
3.4 Clasificación automática	66
Capítulo 4. Particionamiento por medoides PAM.....	70
4.1 Caso de estudio PAM	70
4.1.1 K-medoides en PAM.....	70
4.1.2 Algoritmo de Reubicación Iterativa (RI).....	71
4.1.3 Alcances de K-medoides	72
4.1.3.1 Algoritmo K-medoides.....	73
4.1.4 Modelo PAM	73
4.2 Análisis Computacional.....	77
4.2.1 Resultados de las pruebas de computo	79
Capítulo 5. P-Mediana y Particionamiento	87

5.1 La P-Mediana y el problema de Conglomerado o Particionamiento en la Investigación de Operaciones	87
5.2 Problema de Particionamiento territorial como un problema de la P-mediana.....	88
5.3 Análisis del problema de conglomerado como uno de optimización.....	90
5.4 Enfoque clásico P-mediana	91
5.5 Transformación de Modelo del problema de la P-mediana de programación entera binaria a optimización combinatoria.....	93
5.5.1 <i>El procedimiento para la transformación a través del análisis de un caso.....</i>	95

Prólogo

Conocí a la Dra. Maria Beatriz Bernábe Loranca en el año de 1985, yo era profesor y ella estudiante de la licenciatura en Ciencias de la Computación en la Escuela de Ciencias Físico Matemáticas de la entonces Universidad de Autónoma de Puebla (UAP). Después, en el año de 1995, ella ingresó al cuerpo docente de la Facultad de Ciencias de la Computación de la Benemérita Universidad de Autónoma de Puebla (BUAP), y entonces fuimos colegas. En ese tiempo, la profesora Bernábe realizó su maestría en Ingeniería en calidad en la Universidad Iberoamericana campus Puebla y en el año 2006, ingresó a la División de Estudios de Posgrado de Facultad de Ingeniería de la Universidad Nacional Autónoma de México (UNAM), donde se graduó con honores como doctora en investigación de operaciones.

Desde hace 12 años pertenece al Sistema Nacional de Investigadores. Años más tarde, en 2012, coincidimos en la Universidad Popular Autónoma del Estado de Puebla (UPAEP), yo como estudiante de doctorado y ella en su estancia postdoctoral. Para graduarme, ella fungió como mi directora de tesis doctoral, por lo cual siempre estaré muy agradecido.

Este libro representa un compendio de herramientas cuantitativas que permiten tomar mejores decisiones en sistemas de Particionamiento y agrupamiento aplicados a diseño territorial por medio de modelos y técnicas de investigación de operaciones. Más aún, este libro es consecuencia del trabajo arduo de investigación de varios años, así mismo de los artículos publicados por la Dra. Bernábe en revistas científicas principalmente en el campo de la investigación de operaciones, que entre las publicaciones, se encuentran artículos de Journal Citation Report y algunos con factor de impacto, artículos arbitrados, artículos de índice CONACYT entre otras, donde en la mayoría de estos productos la Dra. Bernábe es la primera autora.

Esta obra se divide en cinco capítulos muy bien diseñados para dar un panorama general del tema, pero atendiendo las particularidades de las aplicaciones. En el capítulo 1 Preliminares, se proporcionan los fundamentos de

matemáticas discretas con un enfoque de Particionamiento y clases de equivalencias con las respectivas referencias. En el capítulo 2 se expone una Introducción al Particionamiento y se muestran los conceptos de Particionamiento, agrupamiento y cluster con un enfoque algorítmico donde el orden de los temas están elegidos cuidando la dificultad creciente del problema, la cual va desde su complejidad computacional. Desde luego también se muestran las referencias entre las cuales se encuentran artículos de la Dra. Bernábe. En el capítulo 3 llamado Propuestas tradicionales del Particionamiento, se describen las condiciones y generalidades de los algoritmos clásicos de Particionamiento para continuar con el capítulo Particionamiento por medoides PAM, en este capítulo 4 se aborda el problema de Particionamiento como un caso de estudio el cual se va desarrollando en torno a los medoides con un enfoque algorítmico y se concluye con una sección para mostrar los resultados de los experimentos computacionales. Puedo decir sin temor a equivocarme, que este es uno de los temas favoritos de la autora y que lo domina en un alto grado. Finalmente, el capítulo 5 nombrado como P-Mediana y Particionamiento, se aplica el modelo del problema de la P-Mediana para buscar soluciones al problema de diseño territorial, entonces se pasa al enfoque clásico del problema de la p-Mediana y se concluye con la transformación del problema de la P-Mediana a un problema de optimización combinatoria.

Quiero felicitar a la Dra. Bernábe por llevar a buen puerto este proyecto de producir su primer libro, pues es mi convicción que en la medida que sigamos produciendo literatura de investigación de operaciones, sean modelos, métodos, teoremas y algoritmos, estamos contribuyendo a la creación de infraestructura que ayudará a las siguientes generaciones a avanzar hacia una cultura de la toma de decisiones basada en la ciencia y coadyuvar a la eficiencia de las operaciones en las organizaciones. Finalmente, con admiración y respeto quiero decir a la Dra. Bernábe que por el contenido y la forma de presentar el conocimiento, este primer libro le proporcionará muchas satisfacciones personales y profesionales, siendo así espero que sea el primero de muchos libros.

Dr. Rogelio González Velázquez

Introducción

Recientemente el volumen de datos generados por distintas actividades ha tenido un incremento además de veloz, importante. A partir de la multiplicidad, origen, variabilidad y número de datos, se persigue que su uso en el análisis permita extraer información valiosa con el fin de evadir incertidumbre y de producir conocimientos que ayuden en la toma de decisiones.

Existen distintas áreas que se encargan del procesamiento y análisis de los datos masivos, dichas disciplinas se apoyan de técnicas enmarcadas en Data Science (DS), Big Data (BD) y Data Mining (DM) entre otros. En particular, una técnica computacional que se identifica como una herramienta de utilidad demandada en tales disciplinas es el agrupamiento de datos.

Por el lado de BD, se puede decir que su propósito fundamental es evaluar grandes volúmenes de datos que superan la capacidad de los procesamientos informáticos, el objetivo principal es analizar en el menor tiempo y de forma eficaz, toda la información utilizando herramientas o softwares para hallar patrones comunes. Aunque DM igualmente analiza grandes volúmenes, esta comprende un conjunto de técnicas y tecnologías que permiten explorar extensas bases de datos de manera automática y busca a la vez, encontrar patrones repetitivos que expliquen el comportamiento de estos datos para obtener una información específica tal que las distintas tareas de una empresa se resuelvan, por ejemplo la optimización.

A veces suelen confundirse los propósitos de algunas áreas cuando se utilizan herramientas que se relacionan mutuamente entre sí, entonces parece apropiado marcar las diferencias, sobre todo cuando un manuscrito se ocupa de discutir una particular técnica/herramienta, tal como lo es este compendio de 1 volumen que incorpora el tema de introducción al Particionamiento a lo largo del documento.

Este es el primer volumen de una serie de 3, el cual se divide en 5 capítulos. El objetivo es proveer material estructurado que permita la comprensión de este tema para estudiantes de distintos grados a partir del nivel licenciatura.

El contenido de este libro, dedicado a una introducción al Particionamiento, es de gran interés cuando se toma como herramienta computacional, en particular, en la solución de problemas de áreas ingenieriles donde el agrupamiento se encuentra implícito; por ejemplo, si se discuten problemas de diseño de zonas o bien de diversos temas de Investigación de Operaciones como los problemas de ruteo, de asignación, agente viajero etc. En este punto, el Particionamiento se puede utilizar como herramienta y a la vez, expresarse como un modelo de optimización, lo cual se trata a partir del capítulo 2. En consecuencia, el lector verá otras bondades del Particionamiento, sobre todo cuando se trata como instrumento algorítmico y herramienta de apoyo para dar respuestas tanto a problemas muy específicos de Investigación de Operaciones (IO), de Optimización o de Inteligencia Artificial (IA) entre otros.

Aunque se encuentra relacionada la IO con la Optimización, la IO se utiliza para programar actividades sujetas a recursos escasos, y para ello, hace uso de técnicas de modelado matemático, análisis estadístico y optimización. Entonces, la IO busca resolver problemas ante una serie de restricciones con criterios de decisión. Para el caso de la teoría de Optimización clásica, se constituye por un conjunto de resultados, métodos analíticos y numéricos enfocados a encontrar e identificar al mejor candidato de entre una colección de alternativas sin tener que enumerar y evaluar explícitamente todas esas alternativas. En matemáticas, ciencias de la computación o economía, Optimización matemática se entiende como la selección del mejor elemento con respecto a algún criterio de un conjunto de unidades disponibles, no obstante, es razonable afirmar que la investigación operativa es uno de los campos de la matemática y que sus bases son la optimización debido a la necesaria maximización o minimización de una función de costo, donde se eligen sistemáticamente valores de entrada que son tomados de un conjunto, y que el valor de la función es procesado computacionalmente.

La generalización de la teoría de la optimización y otras formulaciones, comprende un área grande de las matemáticas aplicadas, pero en general, la optimización significa el descubrimiento de los "mejores valores" para alguna función objetivo con un dominio definido, es decir, se refiere a la capacidad de

hacer o resolver alguna cosa de la manera más eficiente y en el mejor de los casos, utilizando la menor cantidad de recursos.

Considerando lo anterior, se hace énfasis y se justifica que el espacio de este libro, persigue asimilar el beneficio del Particionamiento como herramienta computacional en la solución de problemas de IO y de áreas afines, y que en términos amplios, el agrupamiento se encuentra tácito o explícito como un problema de optimización, por lo tanto, es inevitable mencionar que en las últimas décadas, el término optimización se ha vinculado al mundo de la informática, de las matemáticas, de la gestión de procesos y en la economía, además de la anhelada eficiencia de un sistema, teoría de colas y simulación entre otros. Por otra parte, y en un contexto general, la IA se comprende considerando su origen y la manera en que la inteligencia se codifica trasladándose a aparatos para realizar operaciones u optimizar procesos para la toma de decisiones. Nótese aquí la conexión entre IO e IA.

Hablar de IA ni es fácil y mucho menos corto el asunto. Aquí nos ubicamos para insistir no precisamente en la relación con múltiples áreas, sino con la IO y principalmente con el Particionamiento. Evidentemente, tanto la IA como la IO tienen un cruce en lo que se conoce como la toma de decisiones, pero ello no significa que este punto sea el único que las sitúa en un pedazo del mismo escenario.

Aunque son muchas las áreas que utilizan la IA para dar solución a sus problemas, la IA es y seguirá siendo una rama de las ciencias de la computación y el Particionamiento como herramienta computacional es, fue y será necesaria para resolver problemas no sólo en computación.

Capítulo 1. Preliminares

Agrupamiento informalmente tiene un uso técnico en distintos escenarios. Generalmente agrupamiento en contexto coloquial, es comprendido y utilizado para organizar objetos, formar “colecciones” o grupos siguiendo algún *criterio* con un propósito, y el área de la computación no es la excepción. Entonces, el concepto implícito de organizar tiene significado en el agrupamiento.

El sentido de organizar se sitúa en el proceso de poner algo en *orden*, tal que ese algo puede ser cualquier conjunto de objetos. Hasta este punto, 4 palabras toman importancia para acercarse al tema de agrupamiento: 1) conjunto, 2) orden \approx organizar y 3) grupos. Para propósitos de este capítulo, conjunto, orden y relación de orden preceden de alguna manera al agrupamiento o bien forman parte del proceso (incluyendo la tradicional búsqueda en estructuras de datos).

1.1 Conjuntos

Un conjunto es una colección de elementos diferentes entre ellos, pero poseen propiedades o características comunes de tal modo que sea viable afirmar si un elemento pertenece o no al conjunto *sin importar el orden*. También se pueden establecer relaciones entre los objetos del conjunto o con los elementos de otros conjuntos, lo que hace recordar al álgebra de conjuntos, así como la igualdad, inclusión, conjunto vacío y universal, conjunto potencia y subconjuntos.

Un *subconjunto* significa que dado un conjunto C y una propiedad P de un elemento genérico de C , los elementos de C que poseen esa propiedad forman un nuevo conjunto S llamado subconjunto de C .

En la literatura de teoría de conjuntos, se encuentra a Cantor como aquel que creó una nueva disciplina matemática entre 1874 y 1897: la teoría de

conjuntos, donde se puede hallar toda una serie de teoremas, definiciones, notaciones etcétera (Curry, 2010) ¹.

Algunos de los conjuntos más importantes que se consideran son los siguientes:

$$\mathbb{N} = \{n : n \text{ es un número natural}\} = \{1, 2, 3, \dots\}$$

$$\mathbb{Z} = \{n : n \text{ es un entero}\} = \{\dots, -1, 0, 1, 2, \dots\};$$

$$\mathbb{Q} = \{r : r \text{ es un número racional}\} = \left\{\frac{p}{q} : p, q \in \mathbb{Z} \text{ con } q \neq 0\right\};$$

$$\mathbb{R} = \{x : x \text{ es un número real}\};$$

$$\mathbb{C} = \{z : z \text{ es un número complejo}\}.$$

Podemos encontrar varias relaciones entre conjuntos y realizar operaciones entre ellos. Un conjunto A es un *subconjunto* de B , denotado $A \subset B$ o $B \supset A$, si todo elemento de A también es un elemento de B . Por ejemplo,

$$\{4, 5, 8\} \subset \{2, 3, 4, 5, 6, 7, 8, 9\}$$

y

$$\mathbb{N} \subset \mathbb{Z} \subset \mathbb{Q} \subset \mathbb{R} \subset \mathbb{C}$$

Trivialmente, todo conjunto es subconjunto de sí mismo. Un conjunto B es un *subconjunto propio* de un conjunto A si $B \subset A$ pero $B \neq A$. Si A no es un subconjunto de B , escribimos $A \not\subset B$; por ejemplo, $\{4, 7, 9\} \not\subset \{2, 4, 5, 8, 9\}$. Dos conjuntos son **iguales**, escrito $A = B$, si contienen los mismos elementos. Esto es equivalente $A \subset B$ y $B \subset A$.

Es necesario tener un conjunto sin elementos. Este conjunto se llama *conjunto vacío* y se denota por \emptyset . Notése que el conjunto vacío es un subconjunto de todo conjunto. Para construir conjuntos nuevos a partir de otros conjuntos, es permitido realizar ciertas operaciones:

¹ En 1903, B. Russell demostraría que la teoría de conjuntos de Cantor era inconsistente y cuestionaría la definición de conjunto en la teoría de Cantor (Egner & Denonn, 1961). Posteriormente, se tiene la teoría axiomática de Zermelo, escrita en (Fraenkel et al., 1973). Por otra parte, Von Neumann (Neumann, 1925) y otros sentaron las bases para la teoría de conjuntos actual (Huertas & Manzano, 2002).

La *unión* $A \cup B$ de dos conjuntos A y B se define como

$$A \cup B = \{x : x \in A \text{ o } x \in B\};$$

La *intersección* de A y B se define como

$$A \cap B = \{x : x \in A \text{ y } x \in B\}.$$

Si $A = \{1, 3, 5\}$ y $B = \{1, 2, 3, 9\}$, entonces

$$A \cup B = \{1, 2, 3, 5, 9\} \text{ y } A \cap B = \{1, 3\}.$$

Podemos considerar la unión y la intersección de más de dos conjuntos. En este caso escribimos

$$\bigcup_{i=1}^n A_i = A_1 \cup \dots \cup A_n$$

y

$$\bigcap_{i=1}^n A_i = A_1 \cap \dots \cap A_n$$

para la unión e intersección, respectivamente de los conjuntos A_1, \dots, A_n . También se pueden definir la unión y la intersección de una colección infinita (o arbitraria) de conjuntos. Si $S = \{A_i : i \in I\}$, entonces

$$\bigcup S = \bigcup_{i \in I} A_i = \{x : x \in A_i \text{ para algún } A_i \in S\}$$

y

$$\bigcap S = \bigcap_{i \in I} A_i = \{x : x \in A_i \text{ para todo } A_i \in S\}$$

para la unión e intersección, respectivamente, de los conjuntos en S indexados por I .

Cuando dos conjuntos no tienen elementos en común, se dice que son *disjuntos*; por ejemplo, si P es el conjunto de los enteros pares e I es el conjunto de los enteros impares, entonces P e I son disjuntos. Dos conjuntos A y B son disjuntos precisamente cuando $A \cap B = \emptyset$.

En ocasiones se trabaja con un conjunto fijo U , llamado *conjunto universal*. Para cualquier conjunto $A \subset U$.

Es útil definir el *complemento* de A , denotado por A' , como el conjunto formado por los elementos que no pertenecen a A .

1.1.1 Conjuntos finitos e infinitos

En matemáticas, un conjunto finito es un conjunto que tiene un número finito de elementos donde la cardinalidad o número de elementos es igual a un número natural.

Si un conjunto no es finito, entonces es infinito. Todo conjunto finito es un conjunto numerable, puesto que sus elementos pueden contarse, pero la recíproca es falsa: existen conjuntos numerables que no son finitos (\mathbb{N}) e infinitos no numerables (\mathbb{R}).

A pesar de que este tema es muy conocido, en este volumen se pone especial atención a la diferencia entre la definición de conjunto y el concepto de grupo que posee distintos tipos de objetos con cardinalidad $|A|$. Las descripciones en el agrupamiento precisan la distinción entre clase, conjunto y grupo (formado por algoritmos). El concepto de clase toma un rol interesante en la clasificación (aprendizaje supervisado) y frecuentemente se continúa su estudio después de los conjuntos dado que una clase es un conjunto de conjuntos.

1.2 Clase

En teoría de conjuntos, lógica de clases y sus aplicaciones en matemáticas, una clase es una familia de conjuntos o colección de conjuntos. También pueden pertenecer a la clase objetos matemáticos que no son necesariamente un conjunto, es decir, si se trata de solo objetos o elementos, se asume que cada objeto es un conjunto con un elemento.

Una clase a es un conjunto si es elemento de alguna otra clase, esto es, si existe otra clase B tal que $a \in B$. Es decir, en la teoría de conjuntos, se denomina de manera informal «clase» a toda propiedad expresada por una fórmula de su lenguaje, aun cuando pueda demostrarse que no existe un conjunto que contenga todos los objetos con esa propiedad, de lo contrario es una clase propia. Ejemplos de clases propias son la clase universal, la clase de la paradoja de Russell o la clase de todos los ordinales (Huertas & Manzano, 2002). El concepto de clase aparece al intentar «agrupar» todos los conjuntos (u objetos) que comparten una cierta propiedad (Prisco, 1997).

El orden, como un concepto subyacente en el agrupamiento es obligado no solo en una visión semántica, sino porque es un proceso implícito en la tarea de agrupar. De este modo, se describen dos temas inevitables: relaciones de orden y orden algorítmico.

1.3 Orden en matemáticas discretas

El orden aparece por todas partes, por lo menos, si de matemática y computación se trata. El primer orden que típicamente se estudia en la matemática es el orden \leq de los números naturales. Esta idea es fácilmente extendida a otros conjuntos de números, tal como los enteros y reales (en los reales el concepto intuitivo de orden se rompe). De hecho, la idea de ser mayor o menor que otro número es una de las percepciones básicas de los sistemas de numeración en general. Un ejemplo muy útil es el orden lexicográfico de las palabras en un diccionario.

Los tipos de orden tienen una propiedad especial: cada elemento se puede comparar con cualquier otro elemento, es decir es o mayor, o menor, o igual; aunque esto no siempre es un requisito deseable, por ejemplo, el orden de los subconjuntos de un conjunto. Si un conjunto contiene los elementos de otro conjunto, entonces se puede decir que es mayor o igual, pero existen conjuntos especiales en matemáticas definidos en la teoría del orden. Estos conjuntos son de importancia sustancial en problemas multiobjetivo donde se atienden al menos dos funciones objetivo en compromiso. Si el problema en cuestión debe agrupar distintos tipos de datos con más de un objetivo, una de las estrategias para expresar la solución es utilizando la teoría del orden y formar un frente de Pareto, que se alcanza por los diagramas de Hasse (Bernábe, 2010). La teoría del orden es una rama de la matemática que estudia varias clases de relaciones binarias que capturan la noción intuitiva del orden matemático. La teoría del orden en problemas de optimización multiobjetivo es un tema que debe abordarse con cuidado en otro volumen.

Sea A un conjunto dado no vacío y R una relación binaria definida en A , entonces se dice que R es una relación de orden.

1. Reflexiva: Todo elemento de A esta relacionado consigo mismo. Es decir, $\forall x \in A, xRx$.
2. Antisimétrica: Si dos elementos de A se relación entre sí, entonces ellos son iguales. Es decir, $\forall x, y \in A, xRy, yRx \Rightarrow x = y$.
3. Transitiva: Si un elemento de A esta relacionado con otro, y ese otro a su vez se relaciona con un tercero entonces el primero estará relacionado con este último. Es decir, $\forall x, y, z \in A, xRy, yRz \Rightarrow xRz$ (García, 2001).

1.3.1 Conjuntos parcialmente ordenados

Sea R una relación binaria en un conjunto A . Si R satisface las propiedades reflexiva, antisimétrica y transitiva se dice que R es una relación de orden. En este caso si a y b son elementos de A tales que aRb , entonces se denota por $a \leq b$.

Si se define a un orden como una relación binaria especial, entonces se considera algún conjunto P y una relación binaria \leq en P . Así, \leq es un orden parcial si es reflexiva, antisimétrica, y transitiva, es decir, para todo a, b y c en P , se tiene que

$a \leq a$ (reflexividad)

sí $a \leq b$ y $b \leq c$ entonces $a \leq c$ (transitividad)

sí $a \leq b$ y $b \leq a$ entonces $a = b$ (antisimetría).

Un conjunto con un orden parcial se llama conjunto parcialmente ordenado o poset (del inglés partially ordered set). El término conjunto ordenado a veces se utiliza para los posets, mientras se especifique que no se atribuye otro significado a ninguna otra clase de orden. Comprobando esta propiedad, se ve inmediatamente que el orden de los naturales, enteros, racionales y reales son todos órdenes en el antedicho sentido. Sin embargo, tienen la propiedad adicional de ser total, es decir, $\forall a, b$ en $X, a \leq b$ o $b \leq a$ (totalidad). Este orden se llama orden lineal o cadena.

Dentro de estos órdenes, hay conjuntos que pueden no ser comparables de este modo porque cada uno puede contener algún elemento que no esté presente en el otro (la inclusión de subconjuntos es un orden parcial, en comparación con los órdenes totales).

En un ánimo por los amplios usos prácticos de los órdenes, se pueden definir numerosas clases especiales de conjuntos ordenados, algunas de las cuales han llegado a ser campos matemáticos por sí mismos (Ver figura 1).

La teoría del orden no se restringe a las varias clases de relaciones de orden, sino que también considera funciones apropiadas entre ellas. Un ejemplo de una propiedad de orden teórica viene del análisis donde encontramos con frecuencia a las funciones monótonas (Gierz *et al.*, 2003) (Steven, 2008).

Figura 1

Teoría del orden



1.4 El orden en los algoritmos

El término orden se estudia en al menos otras dos direcciones: a) algoritmos de orden y b) relaciones de orden (visto en 1.3). En ambos, como en los órdenes parciales y totales, subyace la idea de mantener una posición-sucesión de los objetos o situarlos bajo una regla, es decir, se trata de respetar una regla o métrica entre los objetos. Informalmente, un concepto de algoritmo de orden se centra en poner elementos de una lista o un vector en una secuencia dada por una relación de orden. Considerando que ordenar significa poner objetos de una manera específica basándose en un criterio de colocación o comparación,

en computación el ordenamiento de datos tiene distintas formas de hacer el proceso considerando la técnica o algoritmo utilizado y de acuerdo a particularidades específicas del problema. Generalmente, en este orden algorítmico se procesan listas de gran tamaño donde indirectamente también se realizan las *búsquedas*, por tanto, contar con una buena ordenación en combinación con un algoritmo de búsqueda se hace necesario, de tal manera que encontrar la posición de un valor dentro de una lista es importante. Esta afirmación es de valor conveniente en este apartado debido a que en todo agrupamiento se buscan y ordenan los objetos a agrupar. En seguida se discute brevemente el orden desde el enfoque algorítmico.

1.4.1 Algoritmo de ordenamiento

Un algoritmo de ordenamiento asigna elementos de una lista o un vector en una secuencia. Mientras más grande es el arreglo, mayor es el tiempo que se ocupa en el procesamiento que generalmente responde a una relación de orden, es decir, el resultado de salida ha de ser una permutación o reordenamiento de la entrada que satisfaga la relación de orden dada. Las relaciones de orden más usadas son el orden numérico y el orden lexicográfico. Los algoritmos de ordenamiento indirecta o implícitamente deben satisfacer las búsquedas, lo cual complica la solución de orden. Los algoritmos tradicionales de búsqueda son búsqueda lineal, binaria por interpolación y salto de búsqueda. Estos algoritmos tienen características bien definidas y un costo, el cual se mide por el tiempo de ejecución.

A pesar de las bondades para la comprensión del problema de ordenamiento, este tema sigue siendo de interés desde los inicios de la computación, cuya investigación se ha fijado desde el básico BubbleSort hasta los

algoritmos de búsqueda heurística. Estos últimos adolecen por el alto costo computacional para grandes cantidades de datos ².

En cuanto a la complejidad computacional del agrupamiento, este tópico merece un volumen separado porque son necesarias referencias muy puntuales, pruebas algorítmicas con los cálculos que indiquen el costo computacional, el valor de la función objetivo, etc. En estas circunstancias, los métodos aproximados metaheurísticos no pueden excluirse de la discusión sobre complejidad computacional, un tema extenso y apasionante.

Sin duda, los ordenamientos deben ser eficientes, incluso para optimizar su uso con otros algoritmos que demandan listas ordenadas para una ejecución rápida. Los ordenamientos también son útiles para poner datos en forma canónica y para generar resultados legibles por humanos.

Entre los algoritmos tradicionales de búsqueda que comúnmente resuelven procesos implícitos en otros algoritmos (como lo es el agrupamiento), se

² Se entiende en un sentido amplio que la complejidad computacional se encuentra en el peor caso, caso promedio y en el mejor caso en términos de n , donde n que es el tamaño de la lista o arreglo. El orden de una función utiliza la notación $O(n)$ y el mejor comportamiento para ordenar es $O(n \log n)$. Los algoritmos más simples son cuadráticos $O(n^2)$, mientras que los algoritmos que aprovechan la estructura de las claves de ordenamiento pueden ordenar en $O(kn)$ donde k es el tamaño del espacio de claves y dado que el tamaño es conocido a priori, estos algoritmos tienen un desempeño lineal $O(n)$ (Bose & Nelson, 1962) (Martin, 1971). Algunos de los algoritmos más conocidos en estructuras de datos son: 1) Ordenamiento Burbuja (Bubblesort) $O(n^2)$, 2) Ordenamiento por Selección $O(n^2)$, 3) Ordenamiento por Inserción $O(n^2)$, 4) Ordenamiento Rápido (Quicksort) $O(n * \log_2(n))$ (Ayala *et al.*, 2019).

Por otro lado, los métodos de búsqueda heurística tienen otra manera de buscar, generalmente aleatoria y disponen de alguna información sobre la proximidad de cada solución a la solución objetivo. Esta situación permite explorar en primer lugar los caminos más prometedores. Son características de los métodos heurísticos 1) No garantizan que se encuentre una solución, aunque existan soluciones. 2) Si encuentran una solución, no se asegura que ésta tenga las mejores propiedades (que sea de longitud mínima o de costo óptimo). 3) En algunas ocasiones encontrarán una solución aceptablemente buena en un tiempo razonable.

encuentran a búsqueda lineal, binaria por interpolación y jump. La complejidad se puede medir contando comparaciones: a) Mejor caso: el elemento buscado está en el centro, por lo tanto, se hace una sola comparación, b) Peor caso: el elemento buscado está en una esquina. Necesitando $\log_2 n$ cantidad de comparaciones, c) Caso promedio: son $\log_2(n/2)$, entonces, a velocidad de ejecución depende logarítmicamente del tamaño del arreglo (Knuth, 1998).

El parámetro básico y fundamental en el ordenamiento para medir el costo computacional, es el tiempo de ejecución, entendido como la complejidad del algoritmo que no tiene que ver con dificultad, sino con el rendimiento, esto es, se debe saber enumerativamente las veces que el algoritmo procesa comparaciones. Si en una lista de n términos realiza n comparaciones, la complejidad es de orden $O(n)$. Por ejemplo, complejidad constante es $O(1)$, complejidad cuadrática es $O(n^2)$ y complejidad logarítmica es $O(n \log(n))$, entre otros. En bibliografía diversa, resaltan múltiples algoritmos de ordenamiento con su asociada complejidad computacional, pero elegir uno en particular no es una tarea fácil, por ello, 4 preguntas al menos deben plantearse: ¿Qué grado de orden tendrá la información que vas a manejar? ¿Qué cantidad de datos vas a manipular? ¿Qué tipo de datos quieres ordenar? ¿Qué tamaño tienen los registros de tu lista? (Bell, 1958).

Los algoritmos de ordenamiento se pueden clasificar de las siguientes maneras:

1. Según el lugar donde se realice la ordenación:

- Algoritmos de ordenamiento interno: en la memoria del ordenador.
- Algoritmos de ordenamiento externo: en un lugar externo como un disco duro.

2. Por el tiempo que tardan en realizar la ordenación con entradas ya ordenadas o inversamente ordenadas:

- Algoritmos de ordenación natural (ocupa el tiempo mínimo de ejecución cuando la entrada está ordenada).
- Algoritmos de ordenación no natural (tarda lo mínimo cuando la entrada se encuentra inversamente ordenada).

3. Por estabilidad: un ordenamiento estable mantiene el orden relativo que tenían originalmente los elementos con claves iguales. Cuando los elementos son indistinguibles porque cada elemento se ordena por la clave completa, la estabilidad no interesa. Los algoritmos de ordenamiento que no son estables se pueden implementar para que sí lo sean. Una manera de hacer esto es modificar artificialmente la clave de ordenamiento de modo que la posición original en la lista participe del ordenamiento en caso de coincidencia.

1.5 El asunto discreto

Una lectura justa cuando el Particionamiento se elige para agrupar un conjunto de datos, son las relaciones de equivalencia, situación que permite contar con bases teóricas en el momento de modelar un problema específico que requiera al Particionamiento clásico.

Por otra parte, el concepto de clasificación suele ser confuso cuando se presume analogía con el Particionamiento, pero en inteligencia artificial, clasificación es una técnica de aprendizaje supervisado y agrupamiento es no supervisado. Aunque informalmente y dentro de la teoría de conjuntos, una “clasificación” corresponde a una partición en un conjunto, entonces hay una correspondencia biyectiva entre las particiones en un conjunto y las relaciones de equivalencia en el mismo (Muñoz, 2002) (Oubiña, 1974) (Pinzón, 1975).

1.5.1 Relaciones de equivalencia

La igualdad es un concepto fundamental en matemáticas, el cual se puede generalizar por medio de las relaciones de equivalencia y las clases de equivalencia.

El principio de relación de equivalencia sobre un conjunto admite establecer una relación entre los elementos del conjunto que comparten cierta característica o propiedad. Esto permite *reagrupar* dichos elementos en *clases de equivalencia*, es decir, “bloques” de elementos similares y facilita la construcción de nuevos conjuntos añadiendo o anexando (“agregando”), todos los elementos de una

misma “clase” como un solo elemento que los representará y que define la noción de conjunto cociente ³.

Una relación de equivalencia en un conjunto X es una relación $R \subset X \times X$ tal que (1):

$(x, x) \in R \quad \forall x \in X$ (propiedad reflexiva);

$(x, y) \in R$ implica $(y, x) \in R$ (propiedad simétrica);

(x, y) y $(y, z) \in R$ implica $(x, z) \in R$ (propiedad transitiva).

Dada una relación de equivalencia R en un conjunto X , se acostumbra a escribir $x \sim y$, aunque $(x, y) \in R$ es una notación válida, incluso se utiliza la notación $=$, \equiv , o \cong .

1.5.2 Clases de equivalencia

En matemáticas, cuando los elementos de algún conjunto discreto S tienen una clara relación de equivalencia definida en ellos, se puede dividir naturalmente el conjunto S en *clases de equivalencia*. Estas clases de equivalencia se construyen de modo que elementos a y b pertenecen a la misma *clase de equivalencia* si y solo si son equivalentes.

Una relación de equivalencia en un conjunto X también se le llama relación binaria \sim en X que satisface (1).

Con frecuencia, para señalar las propiedades de relación de equivalencia se escribe $a \sim a \quad \forall a \in X$; $a \sim b$ implica $b \sim a \quad \forall a, b \in X$ y si $a \sim b$ y $b \sim c$, entonces $b \sim c \quad \forall a, b, c \in X$.

³ Espacio cociente: Hace referencia a cierta estructura matemática que se deriva de otra en la que se ha definido una relación de equivalencia, es decir, si X es una estructura matemática en el que se define una relación de equivalencia \sim , entonces el espacio cociente $X | \sim$ es la estructura matemática inducida en el conjunto de clases de equivalencia con las operaciones entre clases de equivalencia obtenidas de manera canónica a partir de las correspondientes en X .

Conjunto cociente: Si A es un conjunto y \sim una relación de equivalencia, entonces las clases de equivalencia forman una partición del conjunto A . Las clases de equivalencia de la relación integran entre sí un nuevo conjunto, denominado conjunto cociente y denotado $X | \sim$

Manteniendo esta notación, decimos que la clase de equivalencia de un elemento a se denota comúnmente como $[a]$ o $[a]_{\sim}$ y se define como el conjunto $\{x \in X \mid a \sim x\}$ de elementos que están relacionados con el elemento a por \sim .

El conjunto de todas las clases de equivalencia en X , respecto a una relación de equivalencia R se denota como $X \mid R$ y se llama X módulo R .

Además, la aplicación sobreyectiva $x \rightarrow [x]$ de X a $X \mid R$ que hace corresponder cada elemento a su clase de equivalencia, se llama sobreyección canónica o aplicación de proyección canónica.

Cuando se elige un elemento en cada clase de equivalencia, se define una aplicación inyectiva llamada *sección*. Si esta sección se denota por s , se tiene que $s([c]) = c$ para cada clase de equivalencia c . El elemento $s([c])$ se llama representante de la clase c . Cualquier elemento de una clase se puede elegir como representante de la clase, especificando la sección de manera apropiada.

Cada elemento x de X es un miembro de la clase de equivalencia $[x]$ y cada dos clases de equivalencia $[x]$ e $[y]$ son iguales o disjuntas. Por lo tanto, el conjunto de todas las clases de equivalencia de X forma una partición de X : cada elemento de X pertenece a una sola clase de equivalencia (Pinzón, 1975). Por el contrario, cada partición de X proviene de una relación de equivalencia de esta manera, según la cual $x \sim y$ si y solo si x e y pertenecen al mismo conjunto de la partición (Lipschutz, 1970).

De las propiedades de una relación de equivalencia se deduce que $x \sim y$ si y solo si $[x] = [y]$.

En otras palabras, si \sim es una relación de equivalencia en un conjunto X , y x e y son dos elementos de X , entonces estas declaraciones son equivalentes:

Puede demostrarse a partir de las propiedades de las relaciones de equivalencia que *las clases de equivalencia forman una partición* de X . Adicionalmente, cuando el conjunto X tiene alguna estructura definida con una operación de grupo o en una topología y la relación de equivalencia \sim es

compatible con esta estructura, el conjunto cociente a menudo hereda una estructura similar a la de su conjunto origen.

De acuerdo a lo anterior, existe la posibilidad de proponer una operación bien definida para formar grupos a partir de un conjunto dado X , incluso cuando una relación de equivalencia ha sido definida sobre el conjunto X . Por ejemplo, puede expresarse una métrica entre objetos que concedan asociarse geoméricamente a un elemento distinguido ubicado en el centro de cada elemento de la partición. El número de “clases” puede ser a priori o eventualmente, a través de las clases de equivalencia, se puede determinar el número de clases (grupos). El centro que se discute aquí no puede ser centro de gravedad porque estamos hablando de un espacio discreto, por tanto, el nombre apropiado es centroide.

Evidentemente la combinatoria se inserta en este espacio con el *número de Bell*, llamado así por Eric Temple Bell. El número Bell es el número de particiones de un conjunto de n elementos, o equivalentemente, el número de relaciones de equivalencia en el mismo (Skiena,1990). Comenzando con $B_0 = B_1 = 1$, los primeros números de Bell son $B_0 = B_1 = 1$. Los primeros números de Bell son 1, 1, 2, 5, 15, 52, 203, 877, 4140 21147, 115975 ... (Sloane, 2022).

La n -ésima de estas cifras, B_n , cuenta la cantidad de formas diferentes de dividir un conjunto que tiene exactamente n elementos, o equivalentemente, el número de *relaciones de equivalencia*.

1.6 Partición discreta

El concepto de partición está ligado al de relación de equivalencia: toda relación de equivalencia sobre un conjunto A define una partición de A , y viceversa y cada elemento de la partición corresponde a una clase de equivalencia de la relación.

Recuérdese la definición de partición: Se dice que una familia de subconjuntos no vacíos $\{A_1, A_2, \dots, A_n\}$ es una partición del conjunto A si se satisfacen las siguientes afirmaciones:

- a) $A_i \cap A_j = \emptyset$ para todo $i \neq j$.
- b) $\bigcup_{i=1}^n A_i = A$.

Ahora, si esta definición se ajusta a un algoritmo, se construyen grupos consistentes y claros, pero se presentan distintos problemas en las implementaciones. En particular, cuando se quiere obtener una partición en K clases de un conjunto con n objetos, se está en presencia de un problema combinatorio complejo. Por ejemplo, el número de particiones de un conjunto con 60 elementos en 2 clases es aproximadamente 1018, y para 100 elementos en 5 clases anda por 1068. De hecho, se puede probar que el número $S(n, K)$ de particiones diferentes de un conjunto de n individuos en K clases, cumple la ecuación de recurrencia $S(n, K) = S(n - 1, K - 1) + kS(n - 1, K)$. Esto lleva a que $S(n, K) = \frac{1}{K!} \sum_{i=1}^k (-1)^{K-i} \binom{K}{i} i^n$ (Trejos, 2009).

1.7 El quehacer matemático y computacional de crear grupos a partir de un conjunto

Un aspecto que no ha sido tratado en la literatura pero que es tácito e incluso evidente, se puede enunciar en la aparente contradicción de crear grupos o clases por algún procedimiento en presencia de datos u objetos que se encuentran en un conjunto A , tal escenario constituye una discusión muy nutrida entre los investigadores del área. Tradicionalmente se tienen subconjuntos a partir de A y se opera con ellos de acuerdo al álgebra de conjuntos.

Pero si se tiene un conjunto A y se quieren crear grupos, sería ingenuo revisar la teoría de grupos del álgebra abstracta/moderna y enunciar premisas para formalizar que los grupos generados por algoritmos cumplen propiedades de los grupos como estructuras algebraicas. Esto no es sensato y computacionalmente no es posible porque los grupos que se forman satisfacen ciertas condiciones muy claras, por ejemplo, de partición discreta o de clase en caso de clasificación supervisada.

La teoría de grupos ocupa una posición importante en matemáticas además de que los grupos tienen gran utilidad el área de teoría de códigos, conteo,

biología, química, y la física entre otras, pero esta teoría es independiente y ajena al agrupamiento en ciencias de la computación. Por ejemplo, teniendo un subconjunto finito de los naturales bajo la suma y producto tradicionales no se puede crear un sistema algebraico que tenga la estructura de grupos.

El vocablo agrupar es propio de las ciencias de la computación cuando se desea formar conglomerados (grupos o clusters) respetando reglas dentro de un proceso algorítmico. Bajo las técnicas de agrupamiento conocidas o creadas por el programador, para los grupos que se forman dado un conjunto A , ha de definirse una operación especial de disimilitud o similitud para construir k grupos, donde k puede ser dado apriori o no y cada grupo tiene un centro. En resumen, situándose en la importancia de una *operación* para la formación de grupos, se establecen las siguientes consideraciones:

1) La necesidad de una métrica definida entre todos los elementos del conjunto A tal que los objetos estén relacionados con esta métrica. La medida generalmente es de distancia en R^n , y significa que cada objeto del conjunto original A tiene una clara componente espacial. Por tanto, si A es el conjunto de datos y de ese conjunto se van a procesar los datos para formar “subconjuntos” con el propósito de agrupar, entonces se debe formular una operación especial. Específicamente, si tal operación es una métrica, por ejemplo de disimilitud, de este modo no se generan subconjuntos, se crean grupos bien separados entre sí, con un centroide o representante de grupo y con cohesión interna respecto a la métrica.

2) El procesamiento de los datos u objetos del conjunto A , que se ha dotado de una operación binaria con las características de una métrica, digamos de distancia mínima, induce a un *orden entre los objetos*. Para el caso de R^2 se puede observar directamente en el plano que los objetos de cada grupo se encuentran “alrededor” de su centro, denominado aquí como elemento distinguido y único (centroide). La ubicación de cada objeto en los grupos construidos es importante tan solo por su posición, y al estar agrupados, se concluye que se encuentran ordenados por al menos una métrica, la cual puede ser la distancia de cada objeto a su centro del grupo. Es fácil notar que este orden no es el usual, sin

embargo, pueden ordenarse de manera decreciente con respecto a la métrica que provee un número real del cálculo de la distancia de todos los objetos hacia el centro en cada grupo. En la práctica, estas distancias se almacenan en una matriz simétrica.

De 1) y 2), se confirma que si se desean crear subconjuntos de un conjunto A , entonces no se construyen subconjuntos en un sentido estricto porque son grupos los que se forman. Nótese que en estos grupos se viola la propiedad de que en los conjuntos no importa el orden de los objetos.

Por otra parte, no es correcta una analogía entre cerradura, existencia del inverso y operación binaria de la teoría de grupos con las características de grupos descritas en 1) y 2). En realidad, los grupos son “subconjuntos” que poseen las propiedades 1 y 2 entre otras.

Capítulo 2. Introducción al Particionamiento

2.1 Agrupamiento

La gran cantidad de datos que se generan constantemente requieren de procedimientos y herramientas cada vez más exactas y potentes para su análisis.

La inteligencia artificial IA, tiene distintos alcances, sin embargo, uno de los objetivos es el tratamiento y análisis de datos, por tanto, es usual caracterizar los datos de forma simplificada para un análisis en un espacio de dimensión reducida con el fin de observar los datos con mayor claridad y orden. Por ejemplo, asúmase que es necesario conocer el subconjunto de índices bursátiles internacionales más relevantes, entonces se busca identificar la dinámica de un cierto producto o mercado emergente.

Sin importar el problema, el conflicto no es disponer de datos, los hay en gran abundancia; el desafío es extraer información detrás de los datos, otorgarles un significado y establecer conclusiones. Tal proceso, que se ha descrito informalmente se le conoce como minería de datos (Data Mining) ⁴, que no debe confundirse con Big Data, a pesar de poseen quehaceres en intersección ⁵.

⁴ Data Mining se define como el análisis de datos para encontrar relaciones inadvertidas para descubrir patrones que sean capaces de resumir las grandes cantidades de datos en otras estructuras o formas comprensibles y útiles. La entrada se da normalmente en una tabla y la salida puede tener reglas, estructuras de árbol, gráficos, etc. (Han & Kamber, 2006).

⁵ Big Data (macrodatos o inteligencia de datos) es un término empleado en las tecnologías de la información/comunicación donde se dispone de un conjunto de datos grande y complejo que debe ser tratado con herramientas y aplicaciones no tradicionales de procesamiento y que utilizan procedimientos sofisticados y de software especializado. Las principales características del Big Data son: a) Volumen, Variabilidad, Visibilidad, Velocidad, Veracidad y Valor. (Mayer-Schönberger & Cukier, 2013).

Dentro de los principales retos en el procesamiento de datos destaca el ejercicio de integrarlos cuando proceden de distintas fuentes, posteriormente, ajustarlos en la variabilidad de enfoques que el problema exige con la intención es extraer información más valiosa. Así como en la praxis de las ciencias experimentales es importante el uso de instrumental adecuado con unidades correctas, para cualquier tarea de procesamiento de datos el mecanismo de análisis es casi el mismo, y en la analogía con la instrumentación, destaca el uso de métricas que deben ser revisadas para adaptarlas en la integración de los datos y al tipo de datos que se está tratando con el fin de asociarlos.

Algunas veces es ambiguo distinguir las disciplinas subyacentes que cubren a las técnicas de análisis de datos. Se asume que este conflicto responde a la intersección de distintos tópicos. No obstante, la minería de datos puede considerarse como el campo amplio de investigación que contiene a múltiples áreas, por ejemplo, bases de datos, inteligencia artificial, aprendizaje de máquina, estadística, etc.

Dentro de los algoritmos comunes de estas disciplinas, el agrupamiento se destaca por la gran utilidad y éxito demostrado en segmentación de mercado, diseño territorial, localización, ingeniería, medicina y ciencias sociales entre otras.

2.1.1 Algoritmo de agrupamiento

Atender un problema que exige agrupamiento, implica desafíos, entre ellos, explorar la literatura para hallar distintos métodos candidatos y eventualmente, elegir uno o varios para ajustarlos al problema, lo que implica en muchas ocasiones construir un método propio. El algoritmo más apropiado para un problema particular frecuentemente necesita ser elegido experimentalmente, siempre que no exista una razón matemática para optar por un modelo de grupo que se ajuste convenientemente. Se debería tener en cuenta que un algoritmo que está diseñado para determinado modelo no tiene ninguna posibilidad de obtener buenos resultados en un conjunto de datos con un modelo diferente.

Una serie de pasos en el proceso de agrupamiento que mantiene determinado éxito en los resultados, pueden ser los siguientes:

a) *Criterio de agrupamiento*: Este depende de la interpretación sensata por parte del experto, el tipo de agrupaciones que se utilice estará en función de la base del conjunto de datos, que consiste en una función costo o un conjunto de reglas.

b) *Algoritmo de agrupamiento*: Después de haber adoptado una medida de proximidad y un criterio de agrupamiento, este paso se refiere a la elección de un esquema algorítmico específico que describa la estructura de la agrupación del conjunto de datos.

c) *Validación de los resultados*: Una vez que se han obtenido los resultados del algoritmo de agrupamiento, tenemos que verificar si estos son correctos mediante la realización de pruebas.

d) *Interpretación de los resultados*: En muchos casos, los expertos en el campo de aplicación deben comparar los resultados obtenidos con otros experimentos y analizarlos con el fin de sacar las conclusiones correctas. Pero la experiencia dicta que independientemente del algoritmo de agrupamiento elegido, se ha puesto énfasis en la necesidad de contar con al menos dos fases en el proceso: inicialización y construcción ⁶.

En retrospectiva, el agrupamiento parece tener un caudal en el análisis de grupo, originado en antropología por Driver y Kroeber en 1932 e introducido a psicología por Zubin en 1938 y Robert Tryon en 1939, que también fue utilizado por Cattell al principio de 1943 para clasificación de la personalidad psicológica basada en teoría de rasgos (Bailey, 1994) (Tryon, 1939) (Cattell, 1943).

El primer uso del término moderno *data clustering* (agrupamiento de datos) se remonta a 1954 (Anil, 2012) (Jain, 2010).

Desde 1963 se han publicado diversos libros clásicos sobre agrupamiento. A partir de ahí, creció el interés en esta línea y se desarrollaron miles de algoritmos nuevos y específicos. Interés especial se ha tenido en expresar las restricciones del agrupamiento en un modelo, generalmente de optimización, lo

⁶ Se conoce como método efectivo o algoritmo, al procedimiento para encontrar la solución a un problema mediante la reducción del mismo a un conjunto de reglas.

cual ha dado lugar a analizar el carácter combinatorio cuando de agrupamiento por particiones se trata. Esta dinámica implicó que se propusieron métodos y algoritmos para reducir la complejidad computacional estimada.

Algunos investigadores se han concentrado más en resolver un determinado algoritmo de agrupamiento y pocas veces se detienen en conceptualizar al grupo en esta área.

La idea de un grupo de datos similares resulta incompleta y subjetiva porque la definición de similitud es parte del *específico problema de agrupamiento* a resolver. No lo es así para el *problema general de agrupamiento*, por tanto, la conjetura en este punto es que este es el principal motivo de que existan miles de algoritmos de agrupamiento (Estivill-Castro, 2002).

Aun así, los investigadores usan y formulan diferentes modelos de grupo dando lugar a algoritmos distintos para cada modelo.

La idea de un grupo, cuando es encontrado por algoritmos diferentes, varía significativamente en sus propiedades. Entender estos "modelos de grupo" en el agrupamiento es clave para asimilar las diferencias entre los algoritmos.

Generalmente, los modelos de grupo incluyen: a) Conectividad: El agrupamiento jerárquico representa la conectividad debido a que construye agrupaciones basadas en la distancia de las conexiones. En el jerárquico, los objetos que pertenecen a un grupo hijo también pertenecen al grupo padre, b) Centroide: Los algoritmos particionales exigen para la construcción de grupos que los centroides se generen aleatoriamente. Por ejemplo, en K-means se representa a cada grupo por un solo vector medio. Estos algoritmos basados en centros asocian a través de una métrica y de forma iterativa un punto de un grupo más cercano o similar al centro de su grupo que al centro de cualquier otro grupo. La categoría conocida como "basados en la separación entre grupos" reclama que los miembros de cada grupo sean más cercanos o similares a cualquier punto del grupo que a cualquier punto perteneciente a otro grupo, c) Distribución: En estos algoritmos, los grupos son modelados utilizando distribuciones estadísticas, d) Densidad: Un grupo es una de región densa donde concentra a diversos datos. Los puntos agrupados en ese grupo comparten una propiedad común y que

generalmente es especificada por una distancia. Se considera que el algoritmo representativo de esta categoría es Density-Based Methods DBSCAN, e) Subespacios: Este esquema es exigido en Bi-agrupamiento (conocido como Co-clustering o Two-mode-clustering). Se dice que un tipo de este agrupamiento es el conocido como agrupamiento con solapamiento o alternativo, que contrario al agrupamiento duro, los objetos pueden pertenecer a más de un grupo. En general, aquellos llamados de subespacios son algoritmos que dentro de un único subespacio definido, los grupos pueden o deben solaparse, f) Grafos: Partiendo de un conjunto de datos a agrupar, se utiliza la teoría de grafos para crear un subconjunto de nodos en un grafo tal que cada dos nodos en el subconjunto están conectados por una arista, entonces se puede considerar como un grupo. Generalmente cuando de grafos se habla para formar grupos, la adyacencia es una estrategia para determinar la conexión de un objeto hacia otro, g) Contigüidad: Un algoritmo está basado en contigüidad cuando un punto del grupo es más cercano o similar a un punto o a más puntos de ese grupo que a cualquier otro punto. Este concepto se encuentra muy relacionado con el agrupamiento por grafos donde los vértices representan el objeto o dato a agrupar y cuyas aristas representan la adyacencia o contigüidad.

Los agrupamientos pueden ser señalarse como duro si cada objeto pertenece a un grupo o no y es suave o difuso cuando cada objeto pertenece a cada grupo según un grado de pertenencia a través de una probabilidad.

Existen otras distinciones de agrupamientos, por ejemplo, los que trabajan con partición estricta, que significa que cada objeto pertenece a exactamente un grupo. Una variante de partición estricta son aquellos con ruido, los cuales pueden no pertenecer a ningún grupo ya sea por ruido o anomalías.

Los algoritmos de agrupamiento son categorizados de varias maneras, por ejemplo, por modelo de grupo, eficiencia computacional o velocidad de cómputo, eficacia en el problema específico etcétera. De hecho, suelen proponerse diferentes tipos de grupos aún bajo la premisa de que la definición de grupo no existe al menos en un acuerdo unánime dentro de la comunidad científica y cuya discusión ha llevado al análisis de los grupos obtenidos y dictaminar si son

realmente satisfactorios o no. En este espacio, se considera para algunos autores que análisis de grupos es en realidad más un arte que una ciencia (Von-Luxburg *et al.*, 2012).

En este texto, solo se establecen las bases principales para algoritmos jerárquicos y atención especial a los algoritmos por particiones, con esmero en Particionamiento por medoides. Escribir todos los algoritmos de agrupamiento es un reto prácticamente imposible ya que son cientos de propuestas, tanto en artículos publicados de esta área como de referencias bibliográficas. Aparte, no todos los algoritmos proporcionan “modelos” para sus grupos, pero si es muy acostumbrado hallar patrones en los grupos.

A pesar de contar con mucha literatura y artículos que proporcionan principios para los modelos de agrupamiento y sus correspondientes algoritmos, no es sencillo ramificarlos cuando se habla de diversidad de problemas reales, es decir, existe una gran cantidad de problemas que requieren un agrupamiento propio y por tanto no existe un algoritmo "correcto, exacto o general" (Estivill-Castro, 2002).

El tema de agrupamiento sigue generando más algoritmos con sus propias características, incluso cuando se discute sobre la gran variedad de algoritmos que contrastan sobre la idea conceptual de grupo y en la manera de encontrarlos eficientemente, entre otros aspectos. El agrupamiento entonces no es un quehacer con solución directa y menos lineal, es un proceso iterativo y/o interactivo que incluye la tarea de prueba conocida como ensayo/error. Por ejemplo, en la tarea iterativa, es común ejecutar un algoritmo de agrupamiento y registrar los resultados para ajustar los parámetros.

2.2 Clúster

Importancia adecuada se otorga a la discusión sobre una definición formal o única de “clúster”. En el área, la mayoría de los investigadores coinciden en que no es viable porque cada interpretación dependerá de la aplicación; aun así, agrupamiento (clustering) como método, si tiene una descripción dirigida no solo la

formulación, también a los algoritmos de agrupamiento asociados, por tanto, el método de agrupamiento se debe adaptar o ampliar al problema que se aborda.

Por ejemplo, piénsese en algo coloquial como el proceso de agrupar individuos en una población donde se persigue descubrir la forma en que estos se relacionan y también se busca conocer el modo en que los individuos se reúnen dentro de un mismo grupo, tal que sean más similares entre sí que con los individuos que pertenecen a otra población. Esta cuestión es muy común en distintos problemas e induce a dar una definición muy popular y útil en la comunidad de análisis de datos, la cual es recogida y consecuente de la definición de partición discreta que puede ser extendida en un modelo y permite llegar al agrupamiento tanto en sus variantes de modelación como en los algoritmos correspondientes.

Definición. Dado un conjunto de datos representado por $X = [x_1, x_2, \dots, x_N]$, se define una m -agrupación de X o una partición de X en m “conjuntos”, clústeres o grupos C_1, C_2, \dots, C_m de modo que se cumplan las siguientes condiciones:

- $C_i \neq \Phi, i = 1, \dots, m$
 - $\bigcup_{i=1}^m C_i = X$
 - $C_i \cap C_j = \Phi, i \neq j, i, j = 1 \dots, m$
- (1)

El conjunto de vectores que contiene C_i son “más similares” entre si y “menos similares” a las características de los vectores del otro clúster C_j . La cuantificación de los términos similar-diferente dependerá de los tipos de agrupamientos base de la estructura de X .

Según lo anterior, cada objeto es asignado a un único grupo (“clase”), este es el caso de un clúster de tipo duro (Hard Clustering) (Vicente *et al.*, 2005). El termino clase dentro de agrupamiento es muy frecuente e incluso es “parcialmente” válido, algunos investigadores lo usan indistintamente, pero no en este texto donde se pone acento en la diferencia entre clase y grupo (clase es un

concepto cuidado en aprendizaje supervisado y grupo en la teoría de agrupamiento)⁷.

2.3 Proceso para el Agrupamiento

Los métodos de agrupamiento necesitan una medida o criterio para asegurar que los objetos que formen un grupo sean similares entre ellos y a la vez distantes a los objetos de los otros grupos (ver subsección 1.7). Por ejemplo, las medidas de proximidad son la cuantificación de los términos similar o diferente y sirve para saber cuánto se asemejan o difieren dos objetos. El propósito es garantizar que todas las características seleccionadas contribuyen por igual al cálculo de la medida de proximidad y no hay características que predominen sobre otras. En el agrupamiento de individuos u objetos, la proximidad habitualmente viene expresada en términos de distancias y para el análisis de clúster por variables se recurre a medidas del tipo coeficiente de correlación.

Como puede observarse, la praxis del estudio de los métodos de agrupamiento pone al investigador en un exhaustivo trabajo que se suma a la tarea de resolver al menos los siguientes aspectos: (i) la ambigüedad de las formulaciones; (ii) la selección de criterios en el modelo y (iii) los algoritmos efectivos.

Respecto al punto (ii), los métodos de agrupamiento se pueden abordar como un problema de optimización. Esto es, considerando que el objetivo del agrupamiento es reunir objetos similares bajo una medida dada y además mantener disimilaridad con respecto a los objetos de otros grupos, entonces se expresa como función objetivo a la medida o métrica cuando solo un criterio se optimiza. En el agrupamiento, el cálculo de la medida entre los objetos puede ser consecuente con la función objetivo o bien asociarse a ella. Respecto a las restricciones, estas suelen establecerse de acuerdo al modelo entero-binario

⁷ Cada punto u objeto puede pertenecer a todos los clústeres hasta cierto grado y se le conoce como fuzzy clustering con funciones de pertenencia parcial o total de los objetos hacia las clases (Vicente *et al.*, 2005).

desde el enfoque de investigación de operaciones o bien representar tales restricciones en un modelo combinatorio. Al referirse a la medida de similaridad, se define normalmente por proximidad en un espacio multidimensional.

En el caso de “clasificación por particiones” (Particionamiento), la función de costo consiste en la minimización de una medida que generalmente es la distancia entre el centro de cada grupo y los objetos que le pertenecen (con una medida definida de cercanía). Dentro de las medidas de similaridad, se encuentran medidas para variables numéricas, binarias, nominales, ordinales, escalares no lineales y mixtas. Si las variables son numéricas lineales, una de las distancias más conocidas y a la vez útil en el Particionamiento es la distancia euclidiana que además satisface las propiedades de una métrica ⁸. Para las restricciones, es correcto tomar la definición clásica de partición de la matemática discreta y con ello se persigue imponer una estructura a los clusters ⁹. Pevio al proceso de Particionamiento, la organización de los datos más razonable y conveniente consiste en calcular la distancia entre todos los objetos y almacenar los cálculos en una matriz. Tanto la matriz de distancias como el número grupos son los parámetros de entrada al algoritmo.

La distancia euclídea no es más que la generalización a n dimensiones del teorema de Pitágoras. Si las distancias en sí no son importantes sino sólo su comparación, a menudo se utiliza la distancia euclídea cuadrada, es decir, sin la raíz cuadrada, pues la comparación entre distancias euclídeas cuadradas da los mismos resultados que entre las distancias euclídeas y puede resultar mucho más rápida de calcular ya que la raíz cuadrada es una operación computacionalmente costosa.

En general, teniendo un conjunto de entidades u objetos o datos, el agrupamiento llamado también análisis de conglomerados afronta el problema

⁸ $d(A, B) \geq 0$, 2) $d(A, B) = d(B, A)$: simétrica, 3) $d(A, A) = 0$, 4) $d(A, B) \leq d(A, C) + d(C, B)$ desigualdad triangular (Minkowski, 1953), 5) Si $d(A, B) = 0$, entonces $A = B$.

⁹ Definición de Partición Sea $P_M = \{C_1, \dots, C_n\}$ una partición de O en M clusters. Esto es $C_j \neq \emptyset \forall j$; $C_i \cap C_j = \emptyset \forall i, j \neq i$; $\cup_{j=1}^M C_j = O$.

general de encontrar “subconjuntos” especiales llamados grupos, es decir, las técnicas de agrupamiento se utilizan para dividir un conjunto de datos en grupos con características homogéneas pero bien separados manteniendo cohesión interna. La homogeneidad significa que las entidades en el mismo clúster deben ser similares (Hartigan, 1975).

2.4 Taxonomía del Agrupamiento

Previo a la descripción de los tipos de agrupamiento, es preciso decir que el problema del agrupamiento se observa tanto desde la perspectiva de la estadística paramétrica como de la no paramétrica. En la literatura es normal encontrar ambos enfoques (Roberts, 1997). Estos dos paradigmas de la estadística tienen características bien diferenciadas. En la estadística paramétrica, dada una variable aleatoria X , se asume que esta se encuentra caracterizada por una distribución que depende de un número finito de parámetros $X \approx F_{\theta \in \Theta}$, siendo θ un espacio de parámetros de dimensión finita. La ventaja de este paradigma permite que si el modelo está bien especificado, las tasas de convergencia de los estimadores son más rápidas, y por tanto se necesita una cantidad menor de datos en las tareas de inferencia. Por otro lado, en muchos casos pueden alcanzarse resultados respecto a la optimalidad del procedimiento estadístico empleado, por ejemplo, en problemas de bondad de ajuste con el lema de Neyman Pearson. El problema con estos modelos es su poca flexibilidad para modelar algunas relaciones complejas entre los datos.

La estadística no paramétrica no asume restricciones fuertes acerca del mecanismo generador de los datos ya que se restringen a la continuidad o al carácter Lipschitziano la variable aleatoria (Tsybakov, 2009) ¹⁰. Este paradigma posibilita ajustar modelos altamente flexibles, pero a pesar de esta ventaja, las

¹⁰ En matemática, una función $f: M \rightarrow N$ entre espacios métricos (M, d_M) y (N, d_N) se dice que es lipschitziana (o se dice que satisface una condición de Lipschitz o que es Lipschitz continua) si existe una constante $K > 0$ tal que $d_N(f(x), f(y)) \leq K d_M(x, y), \forall x, y \in M$

tasas de convergencia son muy lentas, siendo necesaria en muchos casos, una gran cantidad de datos para mantener la significancia estadística, como ocurre de manera especialmente notable en los espacios de alta dimensión, donde dichas técnicas no son adecuadas. En estos casos, se deben combinar con técnicas de reducción de la dimensión o selección de variables. La selección de variables es importante en Aprendizaje Automático, Minería de Datos, etc., debido a que en muchos problemas se procesan los datos en forma de vectores multidimensionales representados por variables, aunque no siempre todas las variables contribuyen a la clasificación o análisis de los datos (variables irrelevantes y/o redundantes). En “clasificación” no supervisada, los métodos de selección de variables han sido menos estudiados que los métodos para clasificación supervisada dado que generalmente las “clases” no están disponibles durante los procesos de clasificación y/o selección. En ambos casos, es necesario aplicar métodos de selección variables en el contexto del agrupamiento porque las variables irrelevantes o redundantes pueden afectar drásticamente el resultado de los algoritmos de agrupamiento.

Existen otras publicaciones donde se han intentado agrupar todos los algoritmos de agrupamiento (Xu & Wunsch, 2005) (Firdaus & Uddin, 2015). Por otra parte, Friedman identifica 3 categorías: 1) Algoritmos combinatorios: Se consideran los datos observados sin reparar explícitamente en la distribución de probabilidad subyacente. 2) Modelos de mixtura: Consideran que la muestra se encuentra formada por datos independientes e idénticamente distribuidos de una población descrita por una función de densidad de probabilidad. La función está caracterizada por un modelo paramétrico compuesto por una mixtura de funciones de densidad, tal que cada función de densidad es el mecanismo generador de cada grupo. 3) Buscadores de modas: Desde una perspectiva de la estadística no paramétrica, se intentan identificar las distintas modas que se encuentran en el conjunto de datos. De esta forma, se establece una regla que asigna a que grupo pertenece cada dato en base a la cercana a las distintas modas (Friedman *et al.*, 2001) (Varela, 2018) (Firdaus & Uddin, 2015) (Xu & Wunsch, 2005).

En general, los métodos de análisis de conglomerados más estudiados y utilizados se bifurcan en 2 categorías: 1) agrupamiento jerárquico (llamado también análisis de grupos puntuales y 2) agrupamiento no jerárquico (agrupamiento particional).

2.4.1 Métodos Jerárquicos

El agrupamiento jerárquico proporciona una jerarquía de “clases” que se componen conjuntamente de grupos disjuntos o incluidos uno en el otro ¹¹. Los algoritmos jerárquicos se denominan así porque establecen una jerarquía entre los agrupamientos, es decir generan una sucesión de particiones donde cada partición se obtiene uniendo o dividiendo agrupamientos. Finalmente se elige la mejor partición de la jerarquía de particiones alcanzada, la cual depende del problema (se usa en este caso grupo o partición indistintamente) ¹². Usualmente los agrupamientos que se forman por métodos jerárquicos emplean una representación en estructura de árbol llamada dendrograma y que revela los resultados gráficamente (Hansen *et al.*, 1989). Por otra parte, los métodos de agrupación jerárquica utilizan implícitamente una función objetivo local, a

¹¹ Clase. En teoría de conjuntos, lógica de clases y sus aplicaciones en matemáticas, una clase es una familia de conjuntos o colección de conjuntos (u otros objetos matemáticos) que no necesariamente es un conjunto. El concepto de clase aparece al intentar «agrupar» todos los conjuntos (u objetos) que comparten una cierta propiedad. En la teoría de conjuntos de Zermelo-Fraenkel, se denomina de manera informal «clase» a toda propiedad expresada por una fórmula de su lenguaje, aun cuando pueda demostrarse que no existe un conjunto que contenga todos los objetos con esa propiedad, en cuyo caso se denomina una clase propia. Existen otras teorías, como la teoría de conjuntos de Von Neumann-Bernays-Gödel (NBG), en las que las clases son objetos y puede establecerse una distinción entre ambos tipos de «colecciones de objetos». (Holmes, 2012).

¹² Es importante considerar si las variables han de estandarizarse para que tengan media 0 y desviación estándar 1 antes de calcular la similitud entre observaciones para que cada variable adquiera una importancia equivalente en el clustering jerárquico, sobre todo si las escalas de medida son distintas. Aplicar o no el escalado de variables puede depender del problema en cuestión.

excepción del algoritmo de enlace único que maximiza la división de todas las particiones obtenidas (Johnson, 1967).

Dentro de los algoritmos jerárquicos se distinguen dos tipos: aglomerativos o divisivos. En el primer caso, están basados en distancias y parten de una agrupación inicial en la que cada grupo contiene una sola entidad; en cada iteración se realizan sucesivas fusiones de pares de clústeres cercanos hasta que todas las entidades estén en el mismo cluster o se cumpla un criterio de parada. En otras palabras, en los algoritmos aglomerativos la partición inicial considera a cada objeto como un agrupamiento, posteriormente y de manera iterativa se van uniendo los agrupamientos más similares y se finaliza cuando todos los objetos forman un único agrupamiento. Ejemplos de estos algoritmos son: Single Linkage y Complete linkage (Kaufman & Rousseeuw 2005). La principal ventaja de los algoritmos aglomerativos es su rapidez mientras que los algoritmos divisivos, tienen la ventaja de que parten del conjunto total de datos, asimismo, el proceso de división no tiene por qué seguir hasta que cada elemento forme un único agrupamiento. Una dificultad en estos algoritmos es que suelen ser muy lentos porque inicialmente trabajan con más objetos, lo cual hace que los algoritmos jerárquicos más utilizados sean los aglomerativos. Una excelente revisión de los algoritmos de agrupamiento jerárquicos puede encontrarse en (Gordon, 1987) (Hastie *et al.*, 2009).

En los algoritmos divisivos, se construye una agrupación inicial con todas las entidades u objetos en el mismo clúster tal que para cada iteración dividen por bi-agrupaciones sucesivas al clúster hasta que todas las entidades estén aisladas, una en cada grupo o incluso una condición de paro, es decir la partición inicial considera que todos los objetos forman un único agrupamiento y después se van dividiendo los agrupamientos en dos. El proceso puede seguir hasta que cada objeto conforme un único agrupamiento (Kaufman & Rousseeuw, 2005).

Es claro entonces que los algoritmos jerárquicos con sus propiedades no ofrecen particiones óptimas aún después de varias aglomeraciones o divisiones. El método de Ward de varianza mínima es un caso especial de agrupamiento jerárquico aglomerativo con enfoque de optimización y utiliza un procedimiento

general, el cual exige en cada paso un criterio para la elección del par de clusters a fusionar. El criterio obedece al valor óptimo de una función objetivo, y en este caso es el error de la suma de los cuadrados o varianza. Esta función objetivo podría ser "cualquier función que responda al problema a tratar" (muchos de los procedimientos estándares de agrupamiento están contenidos dentro de esta clase general) (Ward, 1963).

El costo computacional del agrupamiento jerárquico aglomerativo es $O(N^2 \log N)$, donde N es el número de entidades consideradas (Müllner, 2013). Por otro lado, el agrupamiento jerárquico divisivo tiene un costo mayor. Supóngase que la dimensión del espacio de las entidades a clasificar es m , el costo sería $O(N^{m+1} \log N)$ (Xu, 2005).

2.4.2 Métodos no jerárquicos

A diferencia de los jerárquicos, los métodos particionales o no jerárquicos, asumen que dado el número de clústeres se encontrará ese número de grupos optimizando exacta o aproximadamente una función objetivo. Un algoritmo particional divide un conjunto de datos en una única partición, mientras que un algoritmo jerárquico divide un conjunto de datos en una secuencia de particiones anidadas, es decir, realiza varios niveles de agrupamientos y se usa cuando los datos de entrada son de gran dimensión.

Habitualmente se denomina algoritmos particionales a aquellos en los que se agrupan patrones homogéneos a partir de una división inicial de los datos que se refina, comúnmente de forma iterativa. La diferencia entre un algoritmo particional y otro reside en la medida o métrica utilizada para cuantificar el agrupamiento (junto con de la función de actualización de la división o partición). Es decir, en los algoritmos de Particionamiento el conjunto de datos es particionado en un número pre-especificado de grupos o conglomerados k , y luego iterativamente, se van reasignando las observaciones a los conglomerados hasta que algún criterio de parada se satisface (función a optimizar). Esto significa que la función criterio empleada estará orientada a la búsqueda aproximada de la mejor partición mediante la exploración de forma iterativa del conjunto de datos.

Debido a que estos algoritmos optimizan un criterio predefinido o una función objetivo dado un conjunto de datos, los algoritmos por particiones asumen un conocimiento a priori del número de k de clústeres en que debe ser dividido, de tal modo que se alcanza una sola partición formada por grupos (elementos de la partición).

El problema en torno a la implementación de los algoritmos particionales se centra en las siguientes consideraciones: Dado n entidades en un espacio métrico d -dimensional, se debe determinar una partición de los patrones en k grupos, de tal manera que los patrones en un grupo sean más similares entre sí que a las entidades en los otros grupos ¹³.

En general, el procedimiento para realizar este tipo de algoritmos consiste en los siguientes pasos:

- a) Seleccionar k puntos representantes (cada punto representa un grupo de la solución) ¹⁴.
- b) Asignar cada elemento al grupo del representante más cercano, de forma de optimizar un determinado criterio.
- c) Actualizar los k puntos representantes de acuerdo a la composición de cada grupo.
- d) Volver al punto b).

Este ciclo se repite hasta que no se mejore el criterio de optimización. Otro inconveniente en este método es la cantidad gigantesca de particiones, incluso teniendo un pequeño número de datos es un obstáculo formar los grupos. Existen varios algoritmos de este tipo, entre ellos está el k-means, que al ser el más difundido otros algoritmos se basan en método como medoids, que se explica más adelante.

¹³ En algunos problemas, dependiendo de las condiciones, lo que conocemos como datos, individuos o entidades, se suele denominar patrones.

¹⁴ Representantes, puntos, núcleos, centros de gravedad, centroides se usan de manera indistinta en un contexto informal o bien, se formula el término de manera estricta de acuerdo a un problema específico.

De acuerdo a lo anterior, se establece que los métodos particionales (o clasificación por particiones o más precisamente agrupamiento particional), se reducen a buscar una sola partición mediante la optimización de algún criterio. Existen básicamente dos tipos de métodos: 1) Los que fijan a priori el número de grupos que tienen la ventaja de la sencillez y rapidez, mientras que otros tienen la desventaja obvia de buscar el número de grupos que incluso dependen de distintos parámetros que deben ser estimados por el usuario y su implementación no es fácil, sobre todo cuando la experiencia es poca. Estos métodos no jerárquicos también pueden clasificarse en umbral secuencial, umbral paralelo y la división para la optimización (Aldenderfer y Blashfield, 1984).

2.4.3 Métodos no jerárquicos por umbral

En el método del umbral secuencial se selecciona un centro de grupo y a partir de él, se agrupan todos los casos dentro de un umbral (con valor de distancia que se especifica previamente). El método del umbral paralelo es parecido al secuencial excepto en que se seleccionan varios centros de grupo simultáneamente y se agrupan los casos que se encuentren dentro del umbral del centro más próximo.

Existen distintos algoritmos que utilizan el concepto de umbral:

- 1) Basados en la teoría de grafos: triangulación de Delaunay, subgrafos altamente conectados, agrupación mediante núcleos de conectividad, técnica de búsqueda de afinidad etc. (Ferreira *et al.*, 1998) (Bichot & Siarry, 2013) (Romero *et al.*, 2006).
- 2) Métodos basados en densidades: En estos métodos se agrupan objetos mientras su densidad (número de objetos) en la “vecindad” se encuentre dentro de un cierto umbral. DBSCAN (Density-Based Methods), uno de los más representativos, utiliza una noción de agrupaciones basada en la densidad y permite el descubrimiento de agrupaciones de forma arbitraria. La idea básica es que para cada punto de un clúster, la densidad de puntos de datos en el vecindario tiene que superar algún umbral (Ankerst *et al.*, 1999).

La propuesta DENCLUE (sin desglose de las siglas) utiliza una función de densidad local que considera solo los puntos de datos que realmente contribuyen a la función de densidad general (Hinneburg & Keim, 1998).

- 3) El vecino más cercano: A pesar de que en este contexto no se abordan los algoritmos de clasificación supervisada, es útil señalar que el clasificador k vecinos más cercanos ($k - NN$), propuesto por Cover y Hart (Cover & Hart, 1967), es muy utilizado debido a su simplicidad y capacidad para resolver problemas complejos de reconocimiento de patrones. Para el proceso de clasificación, $k - NN$ utiliza un conjunto de ejemplos prototipos de cada clase como conocimiento previo donde cada prototipo está descrito por un conjunto de atributos que lo caracterizan, entonces, $k - NN$ maneja un conjunto de n prototipos previamente clasificados; es decir, para cada prototipo, es conocida a priori la clase a la que pertenece. Este conjunto de n prototipos es llamado conjunto de entrenamiento (T). Dado un nuevo prototipo Q a clasificar, el clasificador $k - NN$ compara Q con cada elemento del conjunto T y encuentra los prototipos más cercanos a Q . Cuando los k vecinos más cercanos se han hallado, la clase del prototipo Q se elige con base en algún criterio, por ejemplo, la votación (Hernández, 2009).

Una variante para fijar la clase de Q que mantiene el uso de los k vecinos más cercanos, es $k - NN$ ponderado por distancias (Atkeson *et al.*, 1997). La diferencia consiste en que los prototipos más cercanos a Q tienen mayor peso en el proceso de votación.

El clasificador $k - NN$ ha sido aplicado en compresión de datos (Gersho & Gray, 1991), bioinformática (Niiijima & Kuhara, 2005), recuperación de información (Deerwester *et al.*, 1990) y reconocimiento de caracteres (Bakamidis, 1993), etc.

2.5 Enfoque de Particionamiento por división para la optimización

El método de división para la optimización difiere de los procedimientos de umbral en que los objetos pueden reasignarse posteriormente a otros grupos con el fin de optimizar un criterio general. En particular, K -medias es el método no

jerárquico por optimización más popular. Otros métodos de Particionamiento han surgido con la idea de superar algunos de los problemas que tiene k –medias, (como lo es la obtención de óptimos locales). Algunos de estos algoritmos por particiones para evitar quedar atrapados en un óptimo local, eligen un medoide como centro de los grupos, por ejemplo, PAM, CLARA, FANNY etcétera, que mantienen el enfoque estadístico, de optimización o difuso (Struyf *et al.*, 1997. Más adelante se encuentran detalles de los medoides.

Respecto a la terminología, se puede hallar en las referencias la expresión “clasificación por particiones” y en tal contexto, el investigador debe asumir que se refiere a Particionamiento. El objetivo aquí se sitúa en comprender el significado preciso de los métodos particionales, tal que los algoritmos generan grupos, es decir, el Particionamiento no construye clases, las clases se atienden, usan o construyen en aprendizaje supervisado, por tanto se insiste en que el Particionamiento se forman grupos de acuerdo a la definición discreta de partición matemática; sin embargo, es posible considerar *clases de grupos* o *grupos de clases* (no definido en este texto), aunque en la comunidad del agrupamiento, minería de datos etc., algunos autores utilizan clase y grupo de manera indistinta y confunde a los lectores que inician su estudio en este tema. Por tanto, lo correcto es tener clases en la clasificación de aprendizaje supervisado y los grupos o conglomerados en agrupamiento tanto particional como jerárquico.

Respecto a la notación matemática de Particionamiento, es común que sea distinta en la literatura, pero se cuida el principio matemático de partición.

La teoría tradicional de los métodos de Particionamiento es fundamentalmente k -means de Forgy, nubes dinámicas de Diday, algoritmo de transferencias de Régnier o k -means de McQueen entre otros (Trejos, 2009).

Los métodos de clustering existentes difieren uno del otro en la forma de estructurar los clusters. Aquellos que encuentran clusters que corresponden a una partición del conjunto de objetos se les conoce como métodos de hard-clustering o clustering particional, siendo el más conocido el algoritmo k -means (Vicente *et al.*, 2005).

2.5.1 Método de *k*-medias y Forgy

Con frecuencia Forgy se confunde el método de las *k*-medias ya que hay dos métodos distintos que son llamados con el mismo nombre.

En general, el método de Forgy, se presenta como sigue:

Sea Ω el conjunto de n individuos a agrupar y supondremos que están descritos por p variables cuantitativas x^1, x^2, \dots, x^p . La presencia de variables cuantitativas tiene sentido para el cálculo de promedios y distancias euclídeas, por lo tanto, también es sensato que cada grupo esté representado por su centro de gravedad, esto es, por un individuo ficticio cuyas coordenadas son los valores promedio de las variables para los individuos pertenecientes al grupo. Este es el caso más simple que frecuentemente se usa junto con la distancia euclídea en este contexto.

Originalmente Forgy en 1965, propuso un primer método de re-asignación que consiste básicamente en la iteración sucesiva, hasta obtener convergencia, de las dos operaciones siguientes (Forgy, 1965):

- Representar una clase por su centro de gravedad, esto es, por su vector de promedios.
- Asignar los objetos a la clase del centro de gravedad más cercano.

A partir de esta propuesta, se formula el método de las *k*-medias, en 4 pasos:

1. Escoger una partición inicial, al azar o con base en algún otro criterio.
2. Calcular los centros de gravedad de la partición.
3. Asignar cada objeto a la clase del centro de gravedad más cercano.
4. Repetir los pasos 2 y 3 mientras las clases en el paso 3 se modifiquen, esto hasta que se obtiene la estabilidad en la partición.

Se ha comprobado estabilidad después de unas pocas iteraciones, pero en la praxis computacional, la elección al azar es una muestra de k objetos iniciales que servirán de núcleos iniciales para después asignar todos los demás objetos al grupo del núcleo más cercano, formándose entonces la partición inicial.

Más tarde MacQueen en 1966, propone un método muy similar donde también se representan las clases por su centro de gravedad, y se examina cada

individuo para asignarlo a la clase más cercana (MacQueen, 1966). La diferencia con el método de Forgy es que inmediatamente después de asignar un individuo a una clase, el centro de ésta es recalculado, mientras que Forgy primero realizaba todas las asignaciones y luego recalculaba los centros (Forgy, 1965). Es claro que el método de McQueen depende del orden en que se presentan los datos. Este método de McQueen ya había sido propuesto en Francia por S. Régnier en 1965 (Régnier, 1983). No obstante, en el transcurso de la búsqueda de una partición de consenso, llamada partición central, variantes del método de Forgy son propuestas en Francia como método de nubes dinámicas por Diday a partir de 1967 (Diday, 1972)

La idea principal de Diday consiste en obtener una partición de Ω en k clases bien agregadas, bien separadas y de intersección vacía. El número k de clases es dado a priori y los datos pueden ser de cualquier naturaleza. Este método fue introducido por en 1972 generalizando el método de k -medias de Forgy. Diday propone que cada clase debe tener una representación (llamada núcleo), y luego se hace una búsqueda iterada de núcleos y de particiones, hasta optimizar un cierto criterio (Diday, 1972).

Es McQueen quien propone el nombre k -means, que se usa hasta la fecha, aun si estos métodos también reciben nombres como nubes dinámicas, centros móviles, o re-asignación (McQueen, 1966).

2.5.2 Método de transferencias

Un segundo tipo de métodos de Particionamiento son los algoritmos del tipo de *transferencias*, originalmente propuestos por Régnier y por MacQueen. Consisten en hacer la transferencia entre una clase y otra, de un único elemento de a la vez, haciendo mejorar algún criterio en cada iteración. Los detalles pueden verse en (Régnier, 1983).

2.5.3 Método de nubes dinámicas

Un esquema llamado de nubes dinámicas es un clásico ejemplo de algoritmos de Particionamiento que fijan a priori el número de grupos. Su

importancia reside en que la mayoría de los algoritmos de Particionamiento se apoyan de las nociones de nubes dinámicas. Estos métodos están basados en el principio que un grupo puede ser representada por algún objeto clave, por ejemplo, un punto promedio, un individuo o grupo de objetos, un conjunto de parámetros, etc. A este representante se le llama comúnmente núcleo ¹⁵. El primer algoritmo de este tipo fue propuesto por Forgy y luego fueron propuestos otros similares (Trejos, 2009) (Forgy, 1965).

La idea general consiste en lo siguiente (a):

- asignar los individuos al núcleo más cercano
- calcular los núcleos con las clases formadas en el paso anterior
- iterar los pasos anteriores hasta obtener estabilidad.

Se parte de una configuración inicial de núcleos, y se puede probar que el método converge a una partición que no mejora el criterio. Dependiendo del contexto y del tipo de núcleo, se define un criterio a ser mejorado.

Considérese $\Omega = \{x_1, \dots, x_n\}$ el conjunto finito de n objetos a clasificar y $k < n$ el número de grupos en los cuales que se desea “clasificar”/agrupar a los objetos. Una partición $P = (C_1, \dots, C_k)$ de Ω en k grupos (C_1, \dots, C_k) está caracterizada por las siguientes condiciones: 1. $\Omega = \bigcup_{i=1}^k C_i$, 2. $C_i \cap C_j = \emptyset \forall i \neq j$.

Teóricamente, se tolera que algunas de las clases C_i sea vacía, pero realmente las particiones $P = (C_1, \dots, C_k)$ que se consideran son particiones Ω en k o menos clases. Sin embargo, las particiones óptimas de acuerdo al criterio de inercia contienen exactamente k clases no vacías (Teorema de Hygens) (Trejos, 2009).

Del mismo modo que en los algoritmos jerárquicos, la similitud es un concepto obligado en el Particionamiento. En efecto, sea P_k el conjunto de todas las particiones $P = (C_1, C_2, \dots, C_k)$ de Ω en k clases o menos. Se quieren obtener buenas particiones, es decir grupos lo más homogéneos que se puedan alcanzar y que estén suficientemente separados, es decir, se buscan particiones que reflejen

¹⁵ Representantes, núcleos, centros de gravedad, centroides.

las relaciones de similitud existentes entre los objetos $x_i \in \Omega$. Cada x_i estará caracterizado por p distintos atributos o variables medidos en una escala numérica y cada objeto x_i es en términos amplios, un vector del espacio euclídeo \mathbb{R}^p , entonces en este espacio la métrica euclídea M (matriz simétrica y definida positiva), sirve para definir el producto interno $\langle x_i | x_j \rangle_M = x_i^t M x_j$ entre objetos y la norma $\|x\|_M^2 = x^t M x$. Cuando los algoritmos se implementan bajo estas consideraciones y buscando convergencia, se asume que $M = \text{Id}$ (métrica euclídea clásica). Esto es, para el caso general M se descompone como $U^t U$ y la transformación $z_i = U x_i$ induce a la métrica euclídea clásica con los nuevos datos z_i .

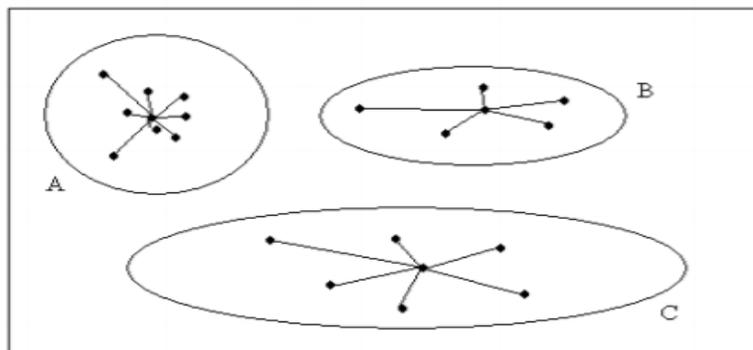
Asociado con cada objeto $x_i \in \Omega$ se tiene el peso de x_i , denotado por w_i , que indica la importancia relativa del objeto x_i en el estudio. Los pesos w_i son todos positivos y su suma es la unidad $\sum_{i=1}^n w_i = 1$.

La calidad de una partición $P = (C_1, \dots, C_k)$ se mide a través de la *inercia inter-clases* $B(P)$, un índice que refleja la intensidad de la separación entre los centros de gravedad de las diversas clases C_j : $B(P) := \sum_{j=1}^k \omega(C_j) \cdot \|g(\Omega) - g(C_j)\|^2$, donde $\omega(C_j)$ es el peso relativo de la clase C_j mientras que $g(\Omega)$ y $g(C_j)$ son los vectores centros de gravedad de Ω y C_j respectivamente.

Sus cálculos corresponden a las siguientes formulas $w(C_j) = \sum_{x_i \in C_j} w_i$, $g(\Omega) = \frac{1}{\sum_{i=1}^n w_i} \sum_{i=1}^n w_i x_i$, $g(C_j) = \frac{1}{\omega(C_j)} \sum_{x_i \in C_j} w_i x_i$ (Piza *et al.*, 1999).

Figura 2

Inercia intra-clase



Como se puede notar en la figura 2, cuanto más pequeña sea la inercia intra-clase mejor es la partición.

En el método general de nubes dinámicas, cada clase estará representada por un núcleo, que será un elemento representativo de los integrantes de la clase.

El algoritmo parte de la idea descrita en (a):

1. Se da una partición inicial de Ω
2. Se calculan los núcleos, mediante una función de representación.
3. Se forma una partición, asignando cada elemento al núcleo más próximo, mediante una función de asignación.
4. Se repiten los pasos 2 y 3 hasta que las clases se estabilicen.

Los núcleos iniciales se eligen de forma aleatoria. En el caso general, se escoge K veces m elementos entre los individuos usando un criterio aditivo del tipo $W(P) = \sum_{k=1}^k \sum_{x_i \in C_k} D(x_i, N_k)$, donde N_k es el núcleo de C_k (formado por m objetos) y D es una medida de disimilitud (por ejemplo, una agregación) entre los objetos x_i y los núcleos N_k (que son conjuntos de objetos). El núcleo N_k se define como el subconjunto de C_k con m elementos que minimice $\sum_{i \in C_k} D(x_i, N_k)$.

Se puede probar que en cada iteración se mejora W y además se converge a una clase estable (Trejos, 2009).

Es claro que el método de k -medias corresponde al método de nubes dinámicas cuando los núcleos son centros de gravedad.

2.6 Suma mínima de distancias al cuadrado en el Particionamiento

Entre muchos criterios utilizados en el análisis de conglomerados (agrupamiento), incluyendo el Particionamiento, la suma mínima de distancias al cuadrado desde los objetos hasta el centroide del clúster al que pertenece es uno de los más utilizados. Este criterio es apropiado tanto para la homogeneidad (similitud) como para la separación (disimilitud), ya que minimizar la suma de cuadrados dentro de los clústeres es equivalente a maximizar la suma de cuadrados entre grupos.

Procedimientos tanto jerárquicos como no jerárquicos han utilizado la suma mínima de cuadrados en la agrupación (MSSC) (Hansen & Aloise, 2009).

2.6.1 Agrupación mínima de suma de cuadrados: Formulación compacta

El Particionamiento busca implícitamente que los grupos generados tengan una forma cercana a un cuerpo geométrico parecido a un polígono regular de muchos lados, o en términos generales, lo más “geométricamente posible”. Esta estrategia se facilita muchas veces cuando los objetos tienen una clara componente geográfica en R^2 y suelen llamarse unidades geográficas. Se supone que la cohesión dentro del grupo cumple implícitamente la compacidad, que de manera informal, se entiende como el proceso de agrupar unidades geográficas de tal manera que estos objetos geográficos queden muy cercanos entre sí y lo más separado que se pueda de otro grupo y al mismo tiempo, procurando que los grupos no tengan formas distorsionadas. Esta informalidad se debe a que esta descripción no responde totalmente al concepto matemático topológico de compacidad, pero es muy útil y válida con resultados comprobados computacionalmente en la solución de muchas aplicaciones de diseño territorial. En este punto, cuando se incorporan los algoritmos de Particionamiento en los problemas de diseño de zonas, por las propiedades del Particionamiento, se optimiza el criterio de distancia mínima y se satisface aproximada e indirectamente el criterio de compacidad (Bernábe, 2010).

Lo anterior, es un asunto discutido ampliamente en el diseño de territorios o diseño de zonas ¹⁶. Los territorios, se particionan para su análisis y se busca que todo elemento de la partición territorial sea al menos geométricamente compacto.

¹⁶ El diseño de territorio (DT) puede ser visto como un problema de agrupación de áreas geográficas pequeñas (áreas básicas) en clusters geográficos más grandes llamados territorios, de tal forma que la agrupación aceptable es aquella última que cumpla con criterios predeterminados del problema que ocupa. Dependiendo del contexto, estos criterios bien pueden ser de motivo económico (promedio de ventas potenciales, trabajo o número de vendedores) o tienen un fondo demográfico (número de habitantes, población votante) (Zoltners & Sinha, 1983).

El uso de medidas o métricas permite conseguir parcial o totalmente la compacidad de acuerdo al problema. Por ejemplo, las distancias, además de ser muy convenientes en este tipo de problemas, permiten acercar los objetos geográficos y así calcular la compacidad geométrica con los ajustes matemáticos necesarios. Es así como se procede a la aplicación de métodos convencionales de agregación (jerárquicos o de partición), en los cuales la restricción de continuidad geográfica se satisface indirectamente a través de la inclusión de las variables geográficas (por ejemplo distancias entre áreas), que en muchas situaciones se encuentran dentro del grupo de variables de clasificación. La inclusión de este tipo de variables favorecerá la creación de regiones espacialmente compactas, lo cual se traduce en la satisfacción de la restricción de continuidad geográfica. La manera en que estas medidas se incorporan en el proceso de agrupamiento determinan la compacidad de cada figura generada por el agrupamiento, alcanzando así la creación de formas geométricas estables y sanciona las formas dispersas, alargadas o retorcidas ¹⁷.

La compacidad en los grupos es muy buscada en el diseño territorial, el cual recurre constantemente a distintas propuestas de agrupamiento para conseguir no solo compacidad, también contigüidad y conexidad en la medida de lo posible (Bernábe-Loranca *et al.*, 2021).

¹⁷ En 1842, el congreso de Estados Unidos creó su primera ley relacionada con el diseño de distritos electorales, donde se dicta que los distritos deben ser conexos para evadir la creación de zonas fragmentadas. En 1871 se suma el concepto de igualdad o equilibrio poblacional, con lo cual se buscaba garantizar el principio “un hombre, un voto”, con el propósito de una representación política justa entre los ciudadanos. Estos dos principios no fueron suficientes para evitar que la construcción de territorios, por ejemplo de zonas electorales, se beneficiara o afectara la representación política de un partido mediante la agrupación distorsionada o dispersa. Entonces en 1901 se añade el concepto de compacidad geométrica para prevenir la creación de zonas con formas alargadas, irregulares o confusas. Actualmente, los principios de conexidad, equilibrio poblacional y compacidad geométrica son considerados como imprescindibles en el diseño de zonas y sus múltiples aplicaciones (Rincón, 2010).

Los problemas de Particionamiento en el diseño de territorios territorial tienen impacto en diversos ámbitos, donde distritación política, distritos escolares, instalaciones de servicios de emergencia y territorios comerciales son las más importantes y citadas en la literatura. Muchos autores formulan criterios geográficos como medidas de adecuación de las soluciones. Los criterios más importantes son la compacidad y la contigüidad (Shirabe, 2005). De acuerdo con (Kalcsics *et al.*, 2005), un territorio (grupo territorial) es geográficamente compacto si tiene forma aproximadamente redonda y no está distorsionado. No obstante, en el contexto de diseño del agrupamiento territorial, no existe una definición rigurosa. En una gran variedad de trabajos se utiliza como medida de compacidad la suma de distancias entre las unidades básicas y el centroide al que están asignadas, modelando el problema como un problema de la p-mediana (La P-mediana se detalla en la última sección) (Díaz *et al.*, 2012). (Hess & Samuels, 1971) (Zoltners & Sinha, 1983).

2.7 Agrupamiento generalizado de suma mínima de cuadrados (MSSC)

El problema MSSC minimum sum-of-squares clustering (MSSC) al poder ser formulado de múltiples maneras, implica que existan diferentes algoritmos para su solución. La MSSC establece que dado un conjunto de n entidades o datos considerados como puntos en el espacio s -dimensional, el conjunto se divida en k grupos tal que la suma de las distancias al cuadrado de cada entidad al centroide de su grupo es mínima. De este modo, y a partir de lo descrito previamente, en los párrafos previos, se plantea la conjetura que de forma indirecta, con la MSSC, se generan grupos compactos.

Sea $O = \{o_1, o_2, \dots, o_n\}$ que denota el conjunto de entidades a ser agrupadas. Tales entidades son puntos en el espacio Euclidiano R^m . Sea $P_M = \{C_1, \dots, C_n\}$ una partición de O en M clusters. Esto es

$$C_j \neq \emptyset \forall j; C_i \cap C_j = \emptyset \forall i, j \neq i; \bigcup_{j=1}^M C_j = O.$$

Introduciendo las variables binarias x_{jk} tal que

$$x_{jk} = \begin{cases} 1 & \text{si la entidad } o_k \text{ pertenece al cluster } C_j \\ 0 & \text{de otro modo} \end{cases}$$

Por tanto, el problema de agrupamiento de suma mínima de cuadrados puede expresarse de la siguiente manera

$$(1) \quad \min \sum_{j=1}^{,elM} \sum_{k=1}^N x_{jk} \|o_k - z_j\|^2$$

$$\text{sujeto a (s. t.) } \sum_{j=1}^M x_{jk} = 1, \quad k = 1, \dots, N,$$

$$x_{jk} \in \{0,1\}, \quad j = 1, \dots, M, k = 1, \dots, N,$$

Donde z_j denota el centroide del cluster C_j .

$$\sum_{k:o_k \in C_j} \|o_k - z_j\|^2 = \sum_{k,l:o_k, o_l \in C_j} \frac{\|o_k - o_l\|^2}{|C_j|},$$

Por lo tanto (1) puede ser escrito como

$$(2) \quad \min \sum_{j=1}^M \frac{\sum_{k=1}^{N-1} \sum_{l=k+1}^N d_{kl} x_{jk} x_{jl}}{\sum_{k=1}^N x_{jk}}$$

$$\text{s. t. } \sum_{j=1}^M x_{jk} = 1, \quad k = 1, \dots, N,$$

$$x_{jk} \in \{0,1\}, \quad j = 1, \dots, M, k = 1, \dots, N,$$

Donde $d_{kl} = \|o_k - o_l\|^2$

$$(3) \quad \sum_{j=1}^M x_{jk} \geq 1,$$

(Merle *et al.*, 2000)

La interpretación de agrupación mínima de suma de cuadrados (MSSC) consiste en que dado un conjunto de n entidades asociadas con puntos en el espacio euclidiano s -dimensional, se divida ese conjunto en k grupos tal que la suma de las distancias al cuadrado de cada entidad al centroide de su grupo sea mínima. En otras palabras, supóngase que se desea maximizar la división de una partición, esto es, la disimilitud mínima entre dos entidades asignadas a dos

grupos diferentes o minimizar el diámetro, es decir, la mayor disimilitud entre un par de entidades en el mismo grupo (Hansen & Delattre, 1978) (Delattre & Hansen, 1980).

Es claro que el criterio frecuente es la suma mínima de distancias euclidianas al cuadrado de cada entidad al centroide del clúster al que pertenece. Entonces particionar n entidades en k clústeres con este criterio se le conoce como agrupación mínima de suma de cuadrados (MSSC) (Aloise, 2009).

Se utilizan muchos criterios diferentes para expresar homogeneidad y/o separación de los conglomerados que se encontrarán (Hansen & Jaumard, 1997).

Por otro lado, y dada la misma semántica matemática que MSSC, para los métodos clásicos de Particionamiento, se acepta por muchos autores que los algoritmos “cómplices” se apoyen de los principios de k-means o nubes dinámicas, dado su esquema sencillo que se basa primero en fijar el número de clases y después, optativamente a partir de una partición inicial, se hacen iteraciones sobre dos pasos: 1) se calculan los núcleos de los grupos y 2) se asignan los objetos al núcleo más cercano. Es claro que se define alguna medida para acercar los objetos a sus núcleos, por ejemplo, al incorporar como función de costo el criterio MSSC, se satisface de algún modo que la distancia o enlace entre los objetos hacia su respectivo núcleo sea mínimo. Este planteamiento se argumenta en la propuesta de Diday; es decir, a pesar de la aparente simplicidad, este algoritmo converge de acuerdo a un teorema similar al de Huygens (Diday *et al.*, 1982). Tal teorema establece que la suma de las distancias al cuadrado desde todos los puntos de un conjunto dado hasta su centroide es igual a la suma de las distancias al cuadrado entre pares de puntos de este conjunto dividido por su cardinalidad (Hartigan, 1974) (Régner, 1983) (Everitt, 1983).

Capítulo 3. Propuestas tradicionales del Particionamiento

3.1 Algoritmo K-means (medias)

K-medias es uno de los algoritmos de tipo particional más divulgados que se coloca al centro de la literatura de los algoritmos de agrupamiento. El éxito de su popularidad reside en la simplicidad para dividir un conjunto de datos en k grupos o clústers conocidos a priori donde cada clúster tiene asociado un centroide (centro geométrico del clúster).

La forma más sencilla del algoritmo k-means consiste de dos procedimientos: La primera es la asignación de objetos a grupos. Un objeto usualmente es asignado a cuya media está más cercano en el sentido. El segundo procedimiento consiste en el cálculo de las nuevas medias del grupo basándose en las asignaciones. El proceso termina cuando no haya más movimientos de un objeto hacia otro grupo.

El objetivo de este procedimiento es disminuir la suma de errores cuadráticos o función objetivo llamada (Sum Squares Error, SSE), es decir, cuando ya no se pueda disminuir más dicha función será el agrupamiento asignado. El problema de estos esquemas es el fracaso cuando los puntos de un grupo se encuentran muy cerca del centroide de otro grupo y cuando los grupos tienen diferentes tamaños y formas.

3.1.1 Procedimiento del Algoritmo

La idea principal del algoritmo k-means consiste en los siguientes pasos (Amoroso y Ávila, 2015):

Inicialización. Se define k centroides, que generalmente son k elementos seleccionados al azar del conjunto de datos o de los obtenidos mediante aplicación de alguna técnica de inicialización, posteriormente se toma cada uno de los elementos del conjunto de datos y se los asignan al grupo con el centroide más próximo.

Proceso iterativo. Mientras los centroides no cambien, se procede a calcular la distancia del centroide de cada grupo y volver a distribuir todos los objetos según el centroide más cercano. El proceso se repite hasta que ya no exista alteración en los grupos, es decir, los k centroides no cambian luego de una iteración y es equivalente a decir que el valor de la función utilizada como criterio de optimización no varía (Yolis, 2003).

Condición de convergencia. Existen varias condiciones de convergencia, una de ellas es converger cuando alcanza un número de iteraciones dado, o converger cuando no existe un intercambio de objetos entre los grupos, o converger cuando la diferencia entre los centroides de dos iteraciones consecutivas es más pequeña que un umbral dado (Pérez, 2007).

Si la condición de convergencia no se satisface, se repiten los pasos anteriores del algoritmo.

La complejidad del algoritmo K-means está dada por $O(n * k * I * d)$ en donde:

- n = número de puntos,
- k = número de Clústers,
- I = número de iteraciones,
- d = número de atributos

Se tiene un problema de tipo No Polinomial (NP) si k no se fija.

Los diferentes tipos de medidas que pueden ser utilizados en este tipo de algoritmo son: la distancia Manhattan, la distancia euclídea y similitud coseno. La medida de distancia euclídea es la más empleada y pragmática.

Se pueden obtener diferentes agrupamientos para diferentes valores de k y medidas de proximidad donde la función objetivo empleada por K-means se denomina suma de errores cuadráticos denotada por SSE o también conocida como sumas residuales de cuadrados (RRS).

Cuando se utiliza la distancia euclídea, SSE se minimiza usando la media aritmética (por cada atributo o variable), pero cuando se emplea la distancia de Manhattan, SSE se minimiza usando la mediana.

Dado el conjunto de dato $X = x_1, x_1, \dots, x_N$ consta de N puntos, se denota la agrupación obtenida, después de aplicar K-means, es decir

$$C = [C_1, C_2, \dots, C_k]$$

El SSE para este agrupamiento está definido por

$$SSE = \sum_{k=1}^k \sum_{x_i \in C_k} \|x_i - c_k\|^2 \quad (1)$$

En donde c_k es el centroide del agrupamiento. El C_k dado por la ecuación (1), donde el objetivo es encontrar la agrupación que minimice el valor de SSE.

$$c_k = \frac{\sum_{x_i \in C_k} x_i}{C_k} \quad (2)$$

El propósito del proceso iterativo y los pasos de actualización del algoritmo K-means, tienen como objetivo el minimizar el valor de SSE para el conjunto dado de centroides.

3.1.2 Minimización de la suma de errores cuadráticos

Como se ha mencionado, el agrupamiento K-means es esencialmente un proceso de optimización por medio de la minimización de la función objetivo conocida como SSE (Pérez, 2007).

Se puede probar matemáticamente la razón de la elección de las medias de los puntos de un conjunto de datos en un agrupamiento, incluso como un prototipo representativo para una agrupación en K-means.

Sea C_k el k-ésimo agrupamiento, x_i es un punto en C_k , y c_k es la media del k-ésimo agrupamiento. Se puede resolver para la característica de C_j que minimiza los SSE para diferenciar el SSE con respecto a C_j y se establece igual a cero, editando la ecuación (1) como:

$$\text{Dado } SSE = \sum_{k=1}^k \sum_{x_i \in C_k} (C_k - X_i)^2 \quad (3)$$

Y aplicando la diferencial a ambos lados de la ecuación

$$\frac{\partial}{\partial C_j} SSE = \frac{\partial}{\partial C_j} \sum_{k=1}^k \sum_{x_i \in C_k} (C_k - X_i)^2 \quad (4)$$

En donde resulta

$$\frac{\partial}{\partial C_j} SSE = \sum_{X_i \in C_j} (C_j - X_i)^2 \quad (5)$$

De lo que se puede obtener

$$\sum_{X_i \in C_j} 2 * (C_j - X_i) = 0 \Rightarrow |C_j| \cdot C_j = \sum_{X_i \in C_j} X_i \Rightarrow C_j = \frac{\sum_{X_i \in C_j} X_i}{|C_j|}$$

Por lo tanto se concluye que la mejor manera representativa de la minimización del SSE de un agrupamiento es la media de puntos de un agrupamiento, debido a que el SSE automáticamente decrece con cada iteración, convergiendo monótonamente a un mínimo local (Pérez, 2007).

K-means no garantiza que los centroides finales obtenidos sean los que minimizan globalmente la función objetivo SSE, pero en la mayoría de los casos, los k-means convergen rápidamente (método de las k-means convergentes) (Dohan *et al.*, 2015). Entonces para conseguir convergencia se propone lo siguiente:

1) Comenzar con una partición inicial de los individuos en clústeres. Si se desea, la partición puede ser construida usando cualquier método de los establecidos para particiones iniciales. 2) Después tomar cada caso sucesivamente y calcular las distancias a todos los centroides de los clusters; si el centroide más próximo no es el del clúster padre de dicho caso, reasignarlo al clúster con centroide más próximo y recalculando los centroides de los clústeres afectados en el proceso. 3) Finalmente, repetir el paso segundo hasta que se obtenga la convergencia, o sea, continuar hasta que un ciclo completo a través de todos los casos y que no proporcione ningún cambio en los miembros de los clústeres.

De acuerdo a lo anterior, el pseudocódigo de K-means con convergencia se expresa como sigue:

3.1.2.1 Algoritmo ejemplo Algoritmo de Agrupamiento K-means

1: Inicializar aleatoriamente las k particiones. Encontrar $C = [C_1, C_2, \dots, C_k]$

2: while no exista cambios en C do

- 3: Clasificar los objetos de acuerdo al C_j más cercano;
 - 4: Recalcular C de acuerdo al actual;
 - 5: end while
- Out: C

Una desventaja de este algoritmo consiste en que el resultado obtenido es dependiente de la selección inicial de los centroides de los clústeres y puede converger a óptimos locales (Park & Jun, 2009).

Por lo tanto, la selección de los centroides iniciales afecta directamente el proceso principal de K-means y al resultado final del proceso. Existen múltiples métodos para la inicialización, como el método propuesto por MacQueen que busca de una manera simple y aleatoria las k semillas. Este es el método más conocido y sencillo (MacQueen, 1966).

3.2 Agrupamiento K-medoids (medoides)

K-medoides es una variación de K-means, por lo tanto, es muy similar. K-medoids, tiene como objetivo encontrar una solución de agrupamiento que minimice una función objetivo predefinida, lo que implica que agrupa el conjunto de n objetos en k grupos conocidos a priori. K-medoids es más robusto al ruido y a valores grandes de los datos en comparación con el K-means ya que minimiza la suma de diferencias por parejas en lugar de la suma de los cuadrados de las distancias euclidianas. El algoritmo K-medoids persigue minimizar el criterio de error absoluto en lugar del SSE, como en el caso de K-means que procede interactivamente hasta que cada objeto representativo sea en realidad el medoide de la agrupación. Este algoritmo utiliza los medoides en lugar de los centroides como el caso de los algoritmos k-means. K-medoids se basa en el objeto más céntrico de un clúster haciéndolo menos sensible a los valores atípicos en contraste con la agrupación K-Means (Villagra *et al.*, 2009).

Un medoide, puede ser definido como un objeto de un grupo cuyo promedio de disimilitud a todos los objetos en grupo es mínima, es decir, que es un punto más céntrico del clúster (Park & Jun, 2009) (Zhang & Couloigner, 2005).

3.2.1 Procedimiento del Algoritmo

El procedimiento del algoritmo empieza por la inicialización de los agrupamientos mediante la selección aleatoria de los k nodos semillas. Todos los nodos se asignan al clúster del nodo de la semilla más cercana (medido en distancia geodésica), esta distancia es un número entero de los pasos entre nodos, muchos nodos semillas pueden estar a la misma distancia de un nodo dado v , en este caso v es asignado aleatoriamente a uno de los nodos semilla más cercanos. El siguiente paso es calcular la proximidad central para cada nodo en cada agrupamiento. La proximidad central de un clúster, es una medida de la distancia de un nodo a los otros nodos en el agrupamiento. Si el nodo i está en la agrupación k y se denota la distancia geodésica entre un nodo i y un nodo j por $d_g(i, j)$, tal que la cercanía del nodo i hacia otro nodo en esta agrupación es definida como el recíproco de la suma de las distancias geodésicas y se da la relación $c_l(i) = \frac{1}{\sum_{j \in \text{cluster } k} d_g(i, j)}$ (Zhang & Couloigner, 2005).

Los nuevos nodos son los que tienen la proximidad central más pequeña en cada grupo. El procedimiento de iteración se da hasta que se obtenga una solución estable, no necesariamente la convergencia, esto es, tomando el caso de distancias geodésicas iguales y debido a la naturaleza aleatoria de la asignación de nodos a las agrupaciones, la solución puede ser débil. Por lo tanto, el algoritmo se considera estable cuando hay un cambio de menos x porcentaje en el número de medoides en el agrupamiento, entonces se sugiere un valor de x entre 1 y 3 (Zhang & Couloigner, 2005).

En el algoritmo de agrupamiento k -medoids, se considera un punto x_i aleatorio que es usado para remplazar un objeto representativo. El siguiente paso es comprobar el cambio de la composición de los puntos que puede ocurrir en una de las siguientes manera: 1) Estos puntos pueden ahora estar más cerca de x_i , (el

nuevo punto representativo), o también pueden estar cerca a todos los otros conjuntos de puntos representativos. 2) El costo de intercambio se calcula como el criterio de error absoluto para k-medoides. 3) Se ejecuta la operación de reasignación, entendida como costo de intercambio, de esta manera contribuye a la función general de costos, entonces el costo es considerado como la distancia geodésica antes descrita.

Los pasos para este algoritmo se plantean a continuación (Park & Jun, 2009) (Rattigan *et al.*, 2007):

1. Inicializar un conjunto de objetos como semillas para los centros de grupo.
2. Se asignan los nodos a la agrupación centro de la agrupación más cercana (usando la distancia geodésica los lazos son resueltos aleatoriamente).
3. Para cada grupo, se calcula un nuevo conjunto de centros de grupo. Estos son los nodos de cada grupo con el valor más bajo de cercanía con la ecuación.
4. Si el algoritmo se ha estabilizado, salir, de lo contrario ir al paso 2 (Webb, 2003).

3.2.2 Pseudocódigo general de K-medoides

- 1: Choose estimaciones iniciales arbitrarias $\theta_j(0)$ para $\theta_j, j = 1, \dots, m$.
- 2: Repeat
- 3: For $i = 1$ to N
- 4: Determinar el representante más cercano, expresar θ_j , for x_i
- 4: Set $b(i) = j$.
- End {For}
- 5: For $j = 1$ to m
- 6: Actualizar Parámetros: Determinar θ_j como la media de los vectores $x_i \in X$ con $b(i) = j$.
- End {For}

7: Hasta que exista cambio en θ_j 's que se produce entre dos iteraciones sucesivas. (Amoroso & Ávila, 2015).

3.3 El agrupamiento desde el enfoque supervisado y no supervisado

El agrupamiento es una técnica de aprendizaje no supervisada. Se parte de un conjunto de n datos no etiquetados y se trata de clasificarlos en uno o más grupos de objetos similares, donde la similitud entre los objetos es frecuentemente definida utilizando alguna medida de distancia en la función objetivo (euclidiana, coseno, Manhattan, etc.). En términos generales, el algoritmo trata de maximizar la similitud inter-clúster (los objetos asignados al mismo clúster son altamente similares) y minimizar la similitud intra-clúster (objetos en diferentes clúster tienen baja similitud entre ellos) (Tan *et al.*, 2005).

Diferentes métodos han sido planteados para realizar procesos de agrupación no supervisada, entre ellos se encuentran los métodos particionales, los cuales dividen un conjunto de datos en k subconjuntos no superpuestos. Entre los principales algoritmos se pueden mencionar k-medias y k-medoides descritos anteriormente, (Kanungo *et al.*, 2002).

Los principales problemas de estos algoritmos residen en que se debe conocer con anterioridad el valor del número de grupos k , e incluso pueden presentar inconvenientes en la identificación de grupos con formas irregulares, un conflicto que se trata de evitar en el Particionamiento al incorporar una medida o métrica que contribuya a que los objetos se encuentren muy cercanos entre sí y hacia su centroide y al mismo tiempo, que los grupos tengan una aproximada forma "poligonal".

El análisis no supervisado es adecuado cuando no se tiene información explícita acerca de la etiqueta de las "clases" a discriminar, lo que induce a cambiar la semántica de clases por grupos. A partir de ahí se les llama métodos agrupamiento no supervisado y se puede decir que los algoritmos relacionados presentan un compromiso entre costo computacional y efectividad bajo la condición de que sus parámetros iniciales hayan sido establecidos correctamente, por ejemplo, datos sobre problemas que puedan manejarse en el plano cartesiano.

Los métodos no supervisados son de interés porque convergen rápidamente y con buen desempeño en caso de que las características cambien poco en el tiempo. Estos métodos consienten adaptar los datos y son manejables cuando se tiene un conjunto grande de muestras siempre que se incorporen algoritmos aproximados como las metaheurísticas. Sin embargo, la solución generada por un sistema de análisis no supervisado puede verse afectada por parámetros iniciales no adecuados y que invariablemente dan lugar a una mala convergencia.

El agrupamiento no supervisado también puede ser llamado agrupamiento o *clustering* cuando se cuenta con buenos algoritmos en el análisis exploratorio de los datos y consecuentemente, ha motivado al desarrollo de diferentes métodos de agrupamiento con los inevitables problemas conocidos: costo computacional, sensibilidad a la inicialización, clases desbalanceadas, convergencia a un óptimo local, etc.

Al principio de esta sección se mencionó la dificultad de elegir un método de agrupamiento adecuado. La no trivialidad de esta tarea responde a distintos factores, entre ellos, la naturaleza de los datos, las condiciones del problema para agrupar los patrones similares que exige la aplicación, entre otros. En este contexto, el desafío es tener un compromiso entre costo computacional y efectividad en la separabilidad de las clases. La situación se complica cuando el problema exige la inclusión de métodos aproximados, que en ocasiones, se opta por utilizar algoritmos híbridos que mejoren la convergencia y reduzcan el costo computacional.

El análisis no supervisado comprende incluso todos los métodos denominados discriminativos en los que no se requiere de un conocimiento a priori de las clases para la clasificación. Por lo general, sólo necesitan de algún parámetro de inicialización como la cantidad de grupos resultantes o algún otro indicio acerca de la partición inicial (prudentemente clases se identifica como grupo). Entonces, la tarea del análisis no supervisado se coloca en la tarea de agrupar patrones homogéneos sin ninguna información acerca de la naturaleza de las “clases” presentes en el conjunto de datos. Por esta razón, el análisis no

supervisado no genera una clasificación automática, más bien produce un subconjunto de datos homogéneos a partir de algún criterio basado en distancias, disimilitudes o medidas estadísticas. De ahí, que el término de clasificación no supervisada se refiere al agrupamiento de los datos en subconjuntos de elementos similares y no algún tipo de clasificación automática. Existen diversas razones por las que los métodos no supervisados son de interés: permiten categorizar elementos, son rentables cuando el etiquetado de un conjunto grande de muestras no es factible, algunos poseen convergencia con buen desempeño (en caso de que las características cambien poco a lo largo del tiempo), etc.

3.4 Clasificación automática

La clasificación automática puede definirse como la acción ejecutada por un sistema artificial sobre un conjunto de elementos para ordenarlos en **clases** o categorías y los elementos a clasificar pueden ser de cualquier tipo. En particular, es la clasificación automática de textos una de las áreas de investigación que ha cobrado mayor importancia en los últimos años, debido principalmente a los grandes volúmenes de textos digitales que se almacenan en bases de datos empresariales, páginas web y redes sociales. La clasificación automática de textos ha estado ligada históricamente al desarrollo de Máquinas de Aprendizaje, una línea de la Inteligencia Artificial y la Inteligencia Computacional que se basa en el desarrollo de algoritmos que “aprenden” o reconocen patrones recurrentes en cada clase a partir de un gran volumen textos de entrada, previamente clasificados por humanos. La clasificación automática de textos es por lo general un proceso supervisado (Baeza & Ribeiro, 1996). Esto significa que requiere de un conjunto de documentos previamente clasificados por expertos humanos que funcionan como entrenamiento para el sistema. Así, un conjunto de documentos clasificado por un humano en cierta categoría, sirven para que el clasificador automático genere una clasificación propia frente a un documento desconocido. El desempeño del clasificador automático dependerá de qué tan similar sea esta clasificación respecto a la humana, lo que se evalúa con matrices de confusión y otras métricas (Sebastiani, 2002).

Tradicionalmente, en el campo de la clasificación, se considera a un elemento como perteneciente a una sola categoría a partir de un conjunto de ellas. Actualmente se utilizan técnicas estadísticas para la clasificación entre las que se pueden incluir modelos de regresión multivariada, clasificación basada en el vecino más cercano, probabilidad de Bayes, árboles de decisión, redes neuronales, aprendizaje basado en reglas y algoritmos de aprendizaje inductivo.

Se ha conceptualizado la clasificación con el paradigma de asociar un elemento con solamente una clase de un conjunto de clases a las que puede pertenecer, observando a esta tarea como una binaria, es decir, el elemento pertenece o no a la clase. No obstante, en los últimos años en aplicaciones relacionadas con *machine learning* e *information retrieval* se ha puesto en manifiesto la importancia de asociar las fuentes de información no solamente con una clase, considerando a este problema no como un problema binario, dado que un valor no binario puede pertenecer a un subconjunto del conjunto de clases, lo que se conoce como multi etiquetado o multi clasificación.

Dentro de *machine learning*, considerado un contexto de aprendizaje supervisado, se cuenta con un conjunto de objetos y un conjunto de clases a las que pertenecen los objetos, entonces se puede definir como clasificación al uso técnicas de aprendizaje supervisado que dado un objeto cualesquiera, este objeto es representado por un vector de características y es asociado con alguna categoría.

Los clasificadores de aprendizaje supervisado utilizan métodos de aprendizaje inductivo, que permiten clasificar información a partir de un conjunto de aprendizaje, entre los más utilizados se encuentran Árboles de decisión, Naïve Bayes, Máquinas de Soporte Vectorial etcétera (Carrera, 2005).

En la clasificación, vista como una técnica basada en IA, se le enseña a la máquina a actuar según los datos etiquetados por los programadores y usuarios. En base a los parámetros de los que estos disponen, los algoritmos de clasificación de IA ya tienen predeterminadas las categorías.

El clustering como forma de aprendizaje no supervisado, no cambia su semántica en IA. En el agrupamiento no hay entrenamiento ni etiquetado. El hecho

de que haya datos que guardan paridad mediante ciertas características que se consideren relevantes (según la complicitad de un elemento respecto a otro), induce a dar una señal al algoritmo que le permitirá asociar un dato a otro, luego es una clasificación válida basada en la observación de patrones descubiertas por el propio algoritmo y la IA.

A veces el agrupamiento de los datos precede a la clasificación de nuevos datos en alguno de los clusters obtenidos en el agrupamiento; por tanto, el agrupamiento y la clasificación de datos están estrechamente relacionados.

Una de las características más importantes de los algoritmos de agrupamiento es que permiten organizar datos que en principio no se sabe si se pueden clasificar, pero evitando criterios subjetivos, con el agrupamiento se produce información muy valiosa a partir de datos desorganizados.

Una característica común a casi todos los algoritmos de agrupamiento es que no son capaces de determinar por sí mismos el número de grupos idóneo, sino que hay que fijarlo de antemano o bien utilizar algún criterio de cohesión para saber cuándo detenerse. En general ello requiere probar con diferentes números de grupos hasta obtener unos resultados adecuados.

En otra perspectiva, la clasificación automática puede definirse como el campo de la matemática aplicada que pretende resolver, mediante ideas, algoritmos y métodos el problema general de la clasificación: dada una colección de objetos, deseamos clasificarlos en clases o grupos bien diferenciados, de acuerdo con las disimilitudes entre los mismos, de forma tal que las clases sean homogéneas internamente. Los métodos tradicionales de clasificación automática, para encontrar una partición de un conjunto de objetos en un número a priori de clases, obtienen soluciones parciales al problema, esto es, encuentran óptimos locales. Dentro de este enfoque de clasificación automática, también se bifurcan las dos familias ya descritas con insistencia a lo largo de este texto: métodos jerárquicos y métodos de Particionamiento (Piza *et al.*, 1999).

Este planteamiento puede confundir a diferentes lectores cuando leen en algún lado que la clasificación automática es la acción de *clasificar en particiones* y al mismo tiempo encuentran que la clasificación es formalmente utilizada en

aprendizaje supervisado. Lo correcto es ser estrictos en los términos y en vocablos pragmáticos, Particionamiento crea grupos que obedecen al concepto de partición discreta, y las clases se apoyan del concepto de clase y pertenencia.

Otros métodos propuestos son basados en procesos bioinspirados y físicos. Entre los basados en procesos bioinspirados, se pueden considerar las bandadas de pájaros a partir de reglas de interacción local (alineamiento, separación, cohesión, similitud). En los métodos inspirados en procesos físicos, se encuentra la técnica de agrupación basada en teoría gravitacional. Dicha técnica es utilizada para realizar procesos de agrupamiento simulando un sistema de gravitación universal tomando cada dato como una partícula expuesta a campos gravitatorios. Una unidad de masa es asociada a cada punto y los puntos son movidos hacia el centro del clúster debido a los campos gravitacionales (Giraldo *et al.*, 2013).

Capítulo 4. Particionamiento por medoides PAM

4.1 Caso de estudio PAM

Partiendo de la definición de Particionamiento clásico y por medoides, se asegura que PAM (Particionamiento Alrededor de los Medoides), pertenece a la categoría de clasificación por particiones (Kaufman & Rousseeuw, 1987).

Para justificar el comportamiento computacional de PAM, y contando con un conjunto de datos con una clara componente geográfica, este apartado se ocupa de obtener particiones ejemplares y exactas para una instancia dada. Las aproximaciones metaheurísticas no se incluyen en este libro.

El problema de Particionamiento consiste en hallar la partición $P \in P_k$ que maximiza la inercia inter-clases $B(P)$, es decir, maximizar la intensidad de la separación entre los centros de los grupos. La complejidad computacional de este problema es del tipo NP-hard. El tamaño combinatorio de este problema puede verse con el siguiente ejemplo. Si denotamos por $S(n, k)$ y B_n el número de particiones de Ω en k clases no vacías y el número total de particiones de Ω respectivamente, entonces por ejemplo $S(60, 2) \approx 0.58 \times 10^{18}$, $S(60, 5) \approx 0.72 \times 10^{40}$, $S(100, 5) \approx 0.66 \times 10^{68}$, mientras que $B_{10} = 115975$, $B_{15} \approx 0.14 \times 10^{10}$ y $B_{40} \approx 0.16 \times 10^{36}$. Asíumase un problema de Particionamiento donde Ω tenga 100 objetos y el número de clases sea $k = 5$, si existiera sobre la Tierra un computador tan veloz capaz de calcular $B(P)$ para cada una de las particiones P de Ω en un tiempo de 10^{-10} segundos en busca de un máximo global, al computador le tomaría algo más de 2×10^{48} siglos en completar el análisis de todas las particiones del problema (Piza *et al.*, 2009).

4.1.1 K-medoides en PAM

Dentro de los algoritmos de Particionamiento, el Particionamiento por medoides, ha sido eficiente para resolver problemas de tamaño no muy grande y por ello, se ha descrito en la literatura desde enfoques diversos.

A diferencia del algoritmo k-means, el método k-medoides utiliza el objeto más céntrico de un clúster para ser el centro del clúster en lugar de tomar el valor medio de los objetos de un clúster. Por tanto el método k-medoides es menos sensible al ruido y a los valores atípicos, pero el precio es un mayor costo computacional. El siguiente algoritmo describe los pasos generales de los algoritmos k-medoides:

4.1.2 Algoritmo de Reubicación Iterativa (RI)

El algoritmo de reubicación iterativa generalizada se integra de 5 pasos:

Entrada: El número de clúster k , y una base de datos que contiene n objetos.

Salida: Un conjunto de k clústeres que minimiza una función de criterio E .

Método:

- 1) elegir arbitrariamente k centros/distribuciones como solución inicial;
- 2) repetir
- 3) (re)calcular la pertenencia de los objetos según la solución actual
- 4) actualizar algunos/todos los centros/distribuciones de los clústeres según a las nuevas pertenencias de los objetos;
- 5) hasta que (no haya cambios en E).

El estado de inicialización k-medoides y variantes es igual a RI en el sentido de la selección aleatoria de los k objetos como centro de los conglomerados (grupos). De las propiedades de k-means, el clustering k-medoides toma el procedimiento de asignar un objeto a su centro más cercano y se resuelve en el paso 4 del algoritmo RI haciendo que el paso 3 sea redundante, por tanto, una mejora de K-medoides dista tanto de RI como de k-medias en que como máximo, se cambiará un centro en el paso 4 para cada iteración. Este cambio de centro debe dar lugar a una disminución de la función criterio, que suele ser la función de error al cuadrado en la minimización de distancias entre objetos y centroides y utiliza implícitamente una medida de distancia (Han *et al.*, 2001).

Para desarrollar el paso 4, un algoritmo de k-medoides llamado PAM es considerado (Kaufman & Rousseeuw, 1987). Entonces se itera a través de todos los k centros de clúster e intenta reemplazar cada uno de ellos con uno de los otros $(n - k)$ objetos. Para cada uno de estos reemplazos, si la función de error cuadrado denotada por E disminuye, el reemplazo tendrá lugar haciendo que se produzca la siguiente iteración como lo hace el algoritmo RI. Si no se encuentra tal reemplazo después de pasar por todos los k clústeres, no habrá ningún cambio en E , esto implica que el algoritmo termina con un óptimo local. Como PAM intenta reemplazar cada uno de los k centros de clúster con uno de los $(n - k)$ objetos y cada uno de estos intentos resulta en $(n - k)$ operaciones para calcular E , la complejidad total de PAM en una iteración es $O(k(n - k)^2)$. Para valores grandes de n , este cálculo se vuelve muy costoso (Han *et al.*, 2001).

Debido a su complejidad, PAM funciona muy bien para conjuntos de datos pequeños, pero no se adapta bien a conjuntos de datos grandes. Esta afirmación justifica que se hayan elegido pruebas de tamaño pequeño para el análisis de esta sección. Cuando se tratan conjuntos de datos más grandes, otras estrategias deben ser desarrolladas. Por ejemplo, el problema de grandes cantidades de datos en particionamiento por medoides, fue expuesto por Kaufman y Rousseeuw, donde desarrollaron un método basado en el muestreo, llamado CLARA (Clustering Large Applications) En esta propuesta, en lugar de tomar en consideración todo el conjunto de datos, se elige una pequeña porción de los datos reales como representante de los mismos. Los medoides se eligen a partir de esta muestra utilizando PAM y la disimilitud media se calcula utilizando todo el conjunto de datos. Si se evalúa un nuevo conjunto de medoides que proporcione una disimilitud menor que la mejor solución anterior, se sustituye la mejor solución por el nuevo conjunto de medoides (Kaufman & Rousseeuw, 1987).

4.1.3 Alcances de K-medoides

Si bien, K-medoides consigue que los medoides sean el punto central de cada clúster, también es se ha comprobado que es más robusto en comparación con k-means. K-medoides encuentra a k como objeto representativo para

minimizar la suma de las disimilitudes de los objetos de datos, mientras que k-means utiliza la suma de las distancias euclidianas al cuadrado para los objetos de datos. El programador debe utilizar sus talentos para diseñar una estrategia para determinar que el objeto medoide se ubique “más al centro”.

Los medoides son el objeto de datos del clúster que se encuentra más centrado, conocidos en un sentido amplio como centroides. Los medoides se seleccionan aleatoriamente de los objetos de datos K_y para formar el clúster K_y y los demás objetos de datos se colocan cerca de los medoides en un clúster. Luego se procesan todos los objetos de datos del clúster para encontrar nuevos medoides de manera repetida para representar el nuevo clúster y que tenga mejor estructura. Después de encontrar los nuevos medoides, se vinculan todos los objetos de datos al clúster. La ubicación de los medoides cambia en consecuencia con cada iteración. Así se forman k_y clústeres que representan n objetos de datos. En este epígrafe, k_y es la notación que se respetó de los autores en (Mark *et al.*, 2002).

4.1.3.1 Algoritmo K-medoides

Entrada: K_y (el número de clústeres), un conjunto de datos que contiene n objetos; Salida: Un conjunto de k_y clústeres.

- Seleccionar aleatoriamente k_y como medoides para n puntos de datos.
- Encuentre los medoides más cercanos calculando la distancia entre los puntos de datos n y los medoides k y asignar los objetos de datos.
- Para cada medoides m y cada punto de datos o asociado a m hacer
- Intercambiar m y calcular el coste total de la configuración
- Seleccionar el medoides con el menor coste de la configuración.
- Si no hay cambios en las asignaciones, repetir los pasos 2 y 3 alternativamente.

4.1.4 Modelo PAM

PAM es un método no jerárquico donde el conjunto de datos es particionado en un número previamente especificado de k conglomerados (grupos

o clústeres), luego iterativamente se asignan las observaciones a los conglomerados hasta que se satisface algún criterio de parada (función a optimizar), por ejemplo, que la suma de cuadrados dentro de los conglomerados sea mínima (Kaufman & Rousseeuw, 1987).

Como ya se ha repetido, la función PAM está basada en la búsqueda de k objetos representativos llamados medoids (medoides) entre los objetos del conjunto de datos. Estos medoids son calculados tal que el total de disimilitud de todos los objetos hacia su medoide más cercano es mínima: es decir, la meta es encontrar un subconjunto $\{m_1 \dots m_k\} \subset \{1 \dots n\}$ que minimiza la función objetivo:

$$\sum_{i=1}^n \min_{t=1, \dots, k} d(i, m_t)$$

Cada objeto es asignado al conglomerado correspondiente del centroide más cercano. Esto es, el objeto i es colocado en el conglomerado v_i cuando el medoide m_{v_i} está más cercano a i que cualquier otro medoide m_w , o

$$d(i, m_{v_i}) \leq d(i, m_w) \text{ para todo } w = 1, \dots, k$$

Actualmente el algoritmo de PAM consta de 2 pasos:

1.- Construir los centroides iniciales

m_1 es el objeto con el más pequeño $\sum_{i=1}^n d(i, m_1)$

.

.

m_k disminuye el objetivo (1) tanto como sea posible

2.- Intercambio

Repetir hasta lograr convergencia y considerar todos los pares de objetos (i, j) con $i \in \{m_1, \dots, m_k\}$ y $j \in \{m_1, \dots, m_k\}$ y hacer el intercambio $i \leftrightarrow j$ para cualquiera que decrezca más el objetivo.

La función objetivo de PAM depende de las disimilitudes entre los objetos, por tanto esta función solamente necesita la matriz de distancias como entrada (Struyf *et al.*, 1997).

Existen numerosos trabajos que detallan la eficiencia de PAM que incluso se han incorporado en modelos de localización-asignación como la P-mediana y comparado con los óptimos de OR-Library (Romero *et al.*, 2019).

No obstante, el objetivo de este espacio es exponer la experiencia computacional del algoritmo PAM (Particionamiento Alrededor de los Medoides) con respecto a la P-Mediana cuando se contrastan sus propiedades para establecer si existe semejanza algorítmica entre las dos propuestas. Entonces, bajo la conjetura de que ambos algoritmos son equivalentes en la función objetivo y la configuración de los grupos que generan, se han escogido una serie de pruebas para comparar los óptimos y centroides: 1) medoides PAM, 2) medianas P-Mediana con Lingo y 3) centroides de PAM con el software R (Robert & Ross, 1993).

El experimento que apoya la hipótesis de equivalencia entre PAM y la P-Mediana es una consecuencia de la observación hecha a distintas pruebas aleatorias al conjunto de unidades geográficas de la Zona Metropolitana del Valle de Toluca (Bernábe, 2010). Por ejemplo, para una instancia específica, la P-Mediana fue utilizada para un encontrar p tamaño de grupos, y del mismo modo, el Particionamiento se usó para hallar k grupos ($k=p$). En el ensayo se confirmó que los costos objetivo de la P-Mediana y del Particionamiento eran los mismos. El problema implicó procesar los datos para diversos tamaños de grupos, y así, con el fin de verificar si para distintas pruebas la función objetivo se mantenía igual tanto para la P-Mediana como para el Particionamiento, se registró el tiempo, la calidad de las soluciones y la estructura/configuración de los grupos. En efecto, se comprobó que tanto PAM como la P-Mediana proporcionan resultados equivalentes.

Dada la importancia de la P-Mediana en este experimento, revisar su modelo se hace necesario, pero en el siguiente volumen, se explica cuidadosamente.

El problema de la P-Mediana considera lo siguiente: Se requiere particionar un conjunto finito de objetos en exactamente p grupos. Cada uno de dichos grupos

estará caracterizado por uno de los objetos, que es seleccionado como la mediana del grupo y el subconjunto de objetos asignado a dicha mediana. Para cada par de objetos se especifica una distancia y se requiere minimizar la suma de distancias entre los objetos y las medianas a las que están asignados:

Sea $N = \{1, \dots, n\}$ el conjunto de objetos.

Para cada par (i, j) , $i \in N$, $j \in N$, donde d_{ij} significa la distancia (similitud) entre los objetos i y j ; y dado el número p que denota el número de grupos, se necesita particionar el conjunto N en p subconjuntos disjuntos, es decir, $N = \bigcup_{k=1}^p N_k$ y $N_r \cap N_s = \emptyset$, para todo $r, s \in \{1, \dots, p\}$, $r \neq s$. A continuación se considera el siguiente modelo de programación matemática para el problema. Se definen las siguientes variables de decisión:

$$y_i = \begin{cases} 1, & \text{si el objeto } i \text{ es seleccionado como mediana,} \\ 0, & \text{en otro caso.} \end{cases}$$

y

$$x_{ij} = \begin{cases} 1, & \text{si el objeto } j \text{ se asigna a la mediana } i, \\ 0, & \text{en otro caso.} \end{cases}$$

El problema se puede modelar de la siguiente manera:

$$\min \quad Z = \sum_{i \in N} \sum_{j \in N} d_{ij} x_{ij} \quad (1)$$

$$\text{sujeto a} \quad \sum_{i \in N} x_{ij} = 1 \quad j \in N \quad (2)$$

$$\sum_{i \in N} y_i = p \quad (3)$$

$$x_{ij} \leq y_i \quad i \in N, j \in N \quad (4)$$

$$x_{ij} \leq \{0,1\} \quad i \in N, j \in N \quad (5)$$

$$y_i \leq \{0,1\} \quad i \in N \quad (6)$$

Las restricciones (2) aseguran que cada objeto se asigna a una de las medianas. La restricción (3) asegura que se seleccionan p objetos como medianas. Finalmente, las restricciones (4) a (6) aseguran que los objetos solo puedan ser asignados a las medianas seleccionadas (Church, 2008).

4.2 Análisis Computacional

Se busca demostrar computacionalmente que bajo mismas condiciones, PAM y la P-Mediana son modelos equivalentes. Se puede comprobar este supuesto si se comparan los resultados en al menos dos propiedades: costo de la función objetivo y configuración de la partición resultante.

Conviene aquí subrayar que el propósito de demostrar la equivalencia entre Pam y la P-Mediana responde a una reflexión personal que rodeaba un problema de localización y diseño de territorio que se podía atender con la P-mediana pero también con el Particionamiento clásico porque se identificaron resultados similares para ambos modelos, de este modo, se examinaron las metodologías más utilizadas en el Particionamiento territorial, donde destacan los modelos de localización-asignación (“location-allocation”) que utilizan a la P-Mediana y los de Particionamiento de conjuntos (“set partitioning”).

En el caso particular de diseño de territorio, el Particionamiento territorial se modela como un problema de la P-Mediana, incluso el Particionamiento se puede resolver como un problema de la P-Mediana o bien, la P-Mediana usa el Particionamiento para su solución (aun cuando los métodos exactos no son una alternativa). Como consecuencia, se dice que cuando dos algoritmos son equivalentes, la decisión de escoger uno de ellos se centra en valorar que algoritmo genera mayor calidad de las soluciones en el menor tiempo de cómputo (aunque sus valores objetivos difieran en milésimas). Entonces, desde ese enfoque y con la finalidad de probar el comportamiento de PAM vs. P-Mediana para determinar su semejanza, se construyó un experimento simple que contrasta 4 aspectos fundamentales y que garantizan computacionalmente la semejanza planteada: 1) función de costo, 2) centroides similares, 3) costo computacional y 4) configuración de los grupos.

Se dice informalmente que PAM crea particiones donde cada grupo de la partición tiene un “centro” llamado medoide, el cual atrae a los objetos que minimizan su distancia hacia el medoide a través de la función de costo. El desafío consiste en retar a PAM con otro algoritmo cuya “semántica algorítmica” genere

los mismos resultados. En este contexto, se escogió el modelo de la P-Mediana, el cual posee un método equivalente si se considera lo siguiente: Se requiere particionar un conjunto finito de objetos en exactamente p grupos. Cada uno de dichos grupos estará caracterizado por uno de los objetos que es seleccionado como la mediana del grupo y el subconjunto de objetos es asignado a dicha mediana. Para cada par de objetos se especifica una distancia y se requiere minimizar la suma de distancias entre los objetos y las medianas a las que están asignados.

De acuerdo a lo anterior, estamos situados en evaluar 2 algoritmos ampliamente citados y conocidos en la literatura para medir-comparar las características que se han reconocido semejantes: costo de la función objetivo y configuración de los grupos. Ambos algoritmos son formulados también como métodos y se les conoce en términos amplios como de localización-asignación y de clasificación por particiones (P-Mediana y PAM respectivamente).

El experimento se ha diseñado vigilando que el costo de cómputo sea razonablemente justo para ser comparado por los dos algoritmos en cualquier equipo de cómputo, incluyendo una portátil. En este punto, se analizaron las limitaciones del software Lingo (utilizado para computar la P-mediana), y que en la versión estudiantil, Lingo genera hasta 50 variables enteras. Ese tamaño de prueba fijó en el experimento el número de datos (369 objetos geográficos). En estas condiciones, se puede probar imparcialmente la P-mediana con Lingo, PAM con el software estadístico R y PAM con un lenguaje de alto nivel usando el mismo tamaño de los grupos en la entrada del algoritmo.

Lingo da solución a la P-Mediana con Branch and Bound (B&B), y por tanto, genera un valor exacto en el resultado de la función objetivo e implícitamente, produce una partición donde el centro de cada partición es la mediana. Respecto a PAM, el algoritmo construye una partición bajo un esquema exhaustivo-combinatorio y se asume que en este ambiente computacional, PAM y P-Mediana son algoritmos equivalentes en los siguientes puntos: 1) utilizan en la entrada de datos una matriz de distancias y el número de k de grupos, 2) generan una partición, 3) usan la misma función de costo, 4) cada centro de los grupos que

componen la partición persiguen que se asocien al centroide los objetos que minimicen su distancia respecto a él. (Básicamente la equivalencia se prueba en el punto 2 y 3, los demás puntos son tácitos).

4.2.1 Resultados de las pruebas de computo

Las tablas 1 a 5, concentran los resultados de las pruebas hechas a la P-Mediana con Lingo, a PAM con un algoritmo de clasificación por particiones combinatorio y PAM con el software R. Se observa que los resultados no son “exactamente iguales”, excepto en la tabla 4. La diferencia “épsilon” en la función objetivo no constituye una razón para afirmar que los algoritmos no sean equivalentes, en realidad la distancia del valor entre los respectivos valores de las funciones de costo se justifica por el cálculo numérico en la aproximación o truncamiento del respectivo software.

Tabla 1

Pruebas de rendimiento con los centros de cada grupo

K	P-Mediana con (B&B) (Lingo)			PAM		
	Costo	Tiempo (Segundos)	Medianas	Costo	Tiempo (Segundos)	Medoides
4	11.16330	1383.21	72, 95, 132, 283	11.16329	0	252, 72, 132, 283
8	5.995200	1941.34	1, 21, 29, 46, 70, 90, 113, 314	5.9951	3	21, 196, 1, 314, 307, 46, 113, 90
12	3.934600	2547.67	27, 46, 63, 82, 91, 111, 128, 142, 155, 257, 267, 322	3.9392	60	296, 123, 149, 129, 27, 63, 181, 91, 267, 257, 108, 142
24	1.721400	2165.05	7, 36, 47, 50, 51, 54, 63, 75, 87, 90, 102, 111, 114, 116, 119, 135, 159, 186, 195, 210, 264, 272, 307, 311	1.7219	75	116, 186, 132, 329, 20, 142, 63, 50, 215, 210, 250, 119, 135, 307, 36, 159, 53, 264, 272, 7, 311, 114, 102, 111,

En la tabla 2, se observan las medianas y los medoides para la P-Mediana y PAM respectivamente. Por ejemplo, para k=4 la mediana=95 y el medoide=252

son los únicos centroides que no coinciden. De la misma manera para $k=24$, las medianas 47, 51, 54, 75, 87, 90 son distintos a los medoides 20, 53, 132, 142, 215, 250, 329; el resto de los objetos son iguales.

Tabla 2

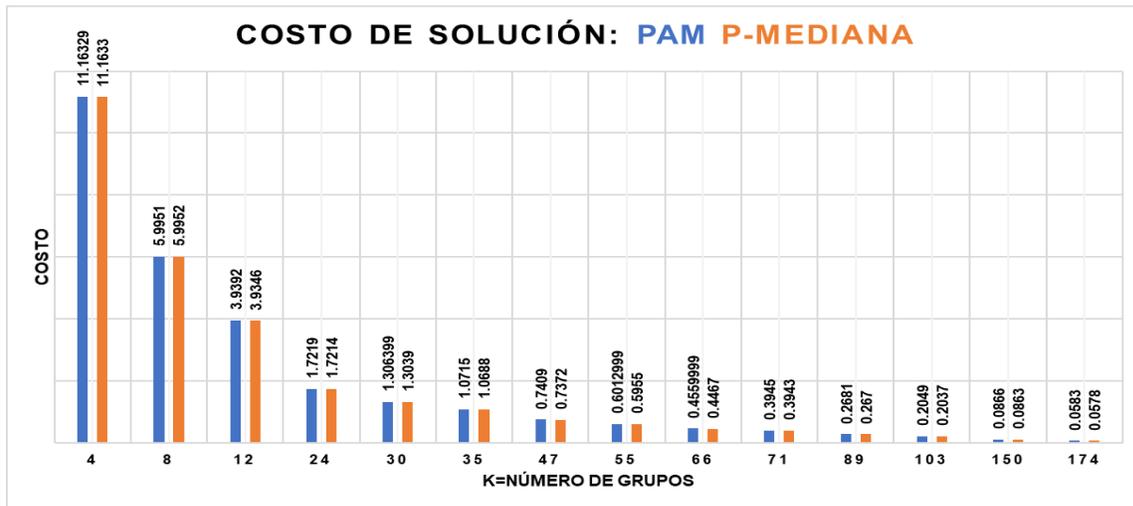
Medianas vs Medoides

k=4		k=8		k=12		k=24	
72	72	1	1	27	27	7	7
95		21	21	46			20
132	132	29		63	63	36	36
	252	46	46	82		47	
283	283	70		91	91	50	50
		90	90		108	51	
		113	113	111			53
			196		123	54	
			307	128		63	63
		314	314		129	75	
				142	142	87	
					149	90	
				155		102	102
					181	111	111
				257	257	114	114
				267	267	116	116
					296	119	119
				322			132
						135	135
							142
						159	159
						186	186
						195	
						210	210
							215
							250
						264	264
						272	272
						307	307
						311	311
							329

En la gráfica de la figura 3, se aprecia la “igualdad” de los óptimos para PAM y la P-Mediana de 4, 8, 12, 24, 30, 35, 47, 55, 66, 71, 89, 103, 150, 174.

Figura 3

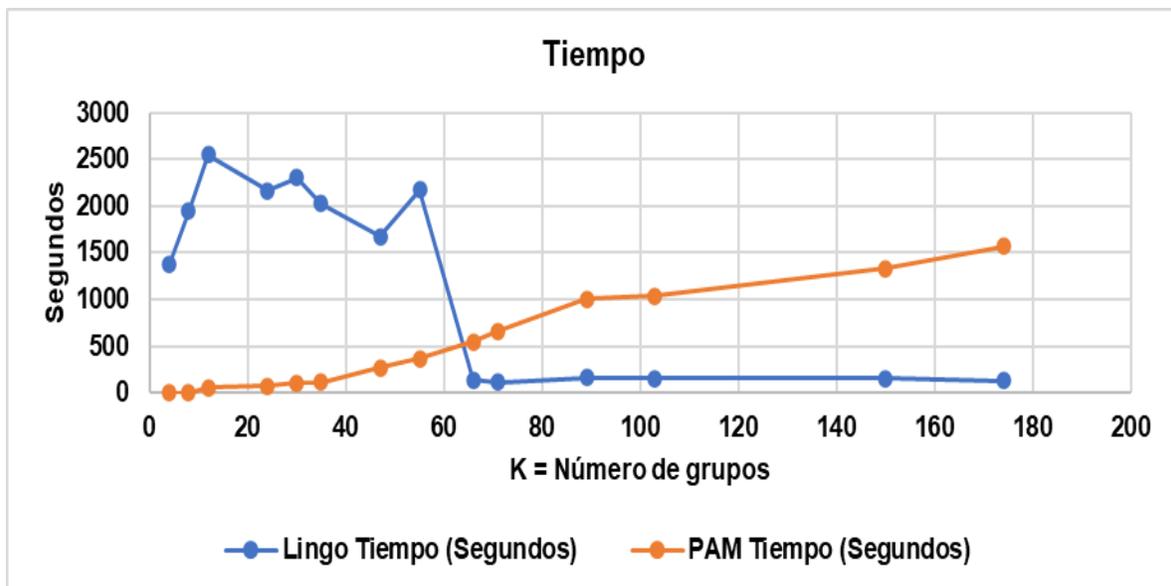
Gráfica de rendimiento entre PAM y P-Mediana



En la figura 4 se observa que el costo computacional de PAM es mejor hasta k=66 grupos con respecto a Ramificación y Acotamiento en Lingo.

Figura 4

Comparación del costo computacional



Los resultados de PAM con un lenguaje de alto nivel y PAM con R vs. P-Mediana en Lingo, tienen una diferencia en los óptimos casi despreciable.

El valor objetivo y la diferencia entre ellos se ha registrado en la siguiente tabla 3.

Tabla 3

Comparación de óptimos PAM con visual, PAM con R y P-mediana con Lingo

K	PAM	Lingo	Lingo-PAM	R	R-PAM
4	11.16329	11.1633	-0.00001	11.1677	0.00441
8	5.9951	5.9952	-0.00010	5.9992	0.0041
12	3.9392	3.9346	0.00460	3.9393	1E-04
24	1.7219	1.7214	0.00050	1.7516	0.0297
30	1.306399	1.3039	0.00250	1.3095	0.003101
35	1.0715	1.0688	0.00270	1.0713	-0.0002
47	0.7409	0.7372	0.00370	0.7538	0.0129
55	0.6012999	0.5955	0.00580	0.601	-0.0002999
66	0.4559999	0.4467	0.00930	0.4483	-0.0076999
71	0.3945	0.3943	0.00020	0.3962	0.0017
89	0.2681	0.267	0.00110	0.2694	0.0013
103	0.2049	0.2037	0.00120	0.205	1E-04
150	0.0866	0.0863	0.00030	0.088	0.0014
174	0.0583	0.0578	0.00050	0.0584	0.0001

En la siguiente tabla se especifica que tanto PAM como la P-Mediana tienen exactamente el mismo costo objetivo para $k=5$:

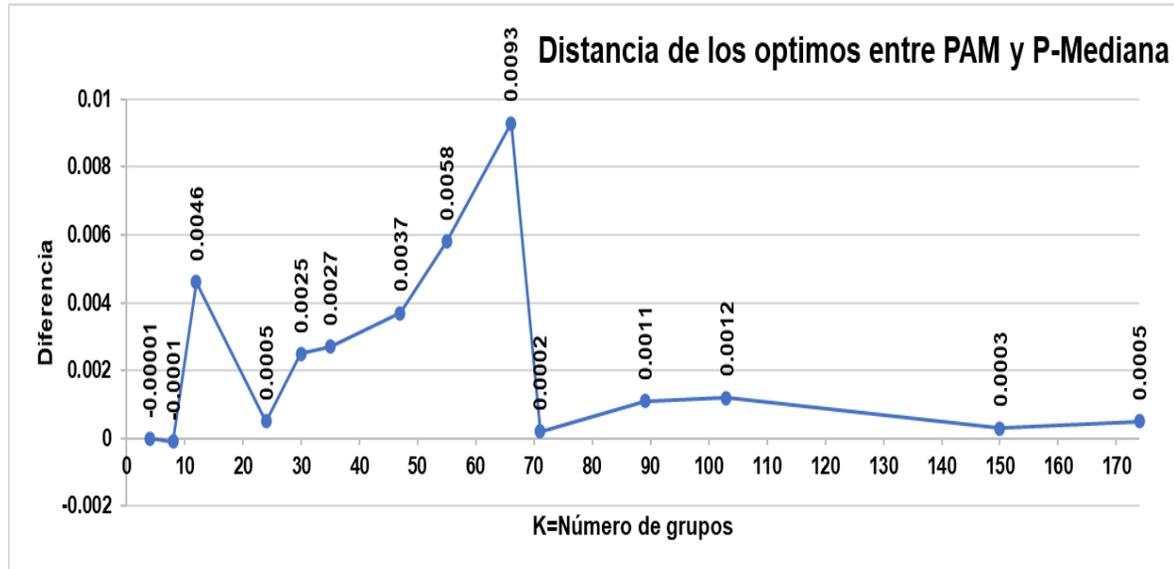
Tabla 4

Comparación entre los dos costos de solución idéntica para $k=5$

PAM	Lingo	PAM = Lingo	Resultado
6.88E-02 (0.06879999)	6.88E-02 (0.0688)	0.068 = 0.068	Iguales

Figura 5

Limites superior e inferior de los costos para PAM y P-Mediana



En la figura 5 es evidente la diferencia mínima de los costos objetivos para los algoritmos evaluados, lo que implica que se mantiene la conjetura de la equivalencia para PAM y P-Mediana.

La tabla 5 muestra el registro de la prueba que alcanzó exactamente el mismo óptimo e incluso los mismos grupos con sus respectivos centros para ambos algoritmos con $k=5$.

Tabla 5

Grupos con sus respectivos centros

K	PAM			P-Mediana	
	Clúster	Medoide	Elementos del grupo	Mediana	Elementos del grupo
1	3		1,2,4	3	1,2,4,5
2	28		15,19,22,23,26,27	14	16,17,20,21,24
5	3	14	16,17,20,21,24	18	11,13
	4	18	11,13	25	6,7,8,9,10,12,29,30
	5	25	5,6,7,8,9,10,12,29,30	28	15,19,22,23,26,27

Después de analizar distintas particiones de la P-Mediana en Lingo y PAM, se tienen conclusiones que quizá muchos investigadores han observado, por ejemplo, la incapacidad de Lingo para generar soluciones a problemas grandes debido a la gran cantidad de variables que genera el algoritmo implícito de Ramificación y Acotamiento. En este documento, la instancia corresponde a 469 objetos geográficos de una zona metropolitana. Con el fin de que Lingo pudiera competir justamente con PAM, se identificó previamente que Lingo solo soporta hasta 369 objetos.

Por otra parte, los algoritmos en PAM son de carácter combinatorio. Sabemos que un problema es de optimización combinatoria (OC) si en ellos las variables de decisión formadas por vectores X sólo admiten valores enteros y su espacio de soluciones factibles Ω está formado por permutaciones o subconjuntos de números naturales.

Respecto a la P-Mediana, tiene ciertas ventajas, por ejemplo, puede ser resuelta con algún optimizador como Lingo que usa el algoritmo de Ramificación y Acotamiento que alcanza una solución exacta, pero las restricciones de los optimizadores para resolver la P-Mediana se enfrentan al número de variables que se generan por el algoritmo del modelo entero-binario. Un dilema que los investigadores tienen en problemas de esta magnitud consiste en optar por dejar a un lado el software de optimización y programar el modelo de la P-Mediana con algún lenguaje de propósito general. Generalmente la P-Mediana es implementada con las características del Particionamiento clásico, siempre que no sea necesario incorporar métodos heurísticos para conseguir una solución aproximada.

En la literatura se encuentran distintas maneras de expresar tanto el Particionamiento como la P-Mediana, incluso con distinta notación pero todas las descripciones tienen la misma intención: crear grupos con un centroide tal que la distancia a cualquier otro objeto del conjunto de datos sea mínima.

Hasta este momento, está muy claro que los centroides en la P-Mediana son las medianas, y tal concepto no debe confundirse con la mediana estadística.

En la P-Mediana, las medianas son la generalización de la mediana conocida comúnmente como una medida estadística descriptiva. Es decir, la P-Mediana usa la mediana geométrica de un conjunto discreto de puntos o datos de una muestra en un espacio euclídeo y es el punto (centro), el que minimiza la suma de las distancias a los puntos de la muestra, por tanto, se generaliza así el concepto de la mediana estadística. El significado de esta mediana se centra en la propiedad de minimizar la suma de distancias para datos unidimensionales y también proporciona una medida de tendencia central en dimensiones superiores. En estas condiciones, es razonable afirmar que la mediana geométrica es un estimador de localización en estadística, incluso también es un indicador estándar en la resolución del problema de localización de instalaciones, donde modela el problema de localizar una instalación para minimizar el costo del transporte (Wolf, 2012).

Formalmente para un conjunto dado de m puntos x_1, x_2, \dots, x_m con cada $x_i \in R^n$ la mediana geométrica se define como $Argmin_{y \in R^n} \sum_{i=1}^m \|x_i - y\|_2$, donde $Argmin$ significa el valor del argumento y que minimiza la suma. En este caso, es el punto y desde donde la suma de todas las distancias euclidianas a x_i es mínima.

Esta afirmación defiende que la mediana geométrica sea la parte sustancial del modelo de la P-Mediana y de forma análoga, que se ajuste como el medoide en el Particionamiento, que desde luego se suman a las restricciones; esto es, recuérdese que una partición de un conjunto A está formada por los subconjuntos A_1, A_2, \dots, A_n los cuales deben cumplir:

- Que la unión de todos los subconjuntos sea igual al conjunto dado $A_1 \cup A_2 \cup \dots \cup A_n = A$
- Que todos los subconjuntos sean disjuntos entre sí y que ningún subconjunto sea vacío.

Estas condiciones implican que el Particionamiento para un conjunto de datos se plantea matemáticamente como la P-Mediana y viceversa, en otras palabras, ambos modelos comparten que la función objetivo se comporte de la misma manera. Los medoides de PAM, al ser centros de grupos, permiten al investigador elegir a su criterio la mejor manera de colocar los medoides en el

centro de los clústeres y así, coincide con la definición de mediana geométrica, que es el centro de los grupos en el modelo de la P-mediana. Finalmente los centros de ambos algoritmos concuerdan (medianas=medoides).

Revisando los costos de los resultados que arrojaron los métodos presentados, estos valores son iguales con una discrepancia muy pobre que se justifica por la precisión de las cifras significativas aritméticas en el truncamiento y aproximación. Este conflicto se puede resolver en la implementación de PAM con algún método numérico como Newton-Fourier, Bisección o Punto fijo, por ejemplo; pero en Lingo, desconocemos la manera en que aproximan o truncan los decimales en la función de costo. Entonces se afirma que ahí habita la diferencia del ϵ entre los valores de las funciones objetivo de PAM y P-Mediana en Lingo. Sin embargo, las normas IEEE en el estándar 754, dictan que los softwares comerciales definen 3 formatos de punto flotante (32, 64, y 128 bits), por tanto, se puede arreglar el problema de aproximación en PAM y de este modo conseguir que los valores objetivos sean exactamente iguales (Institute of Electrical and Electronics Engineers (IEEE), 2019).

En relación a la diferencia de los centros de los grupos, es más sencillo explicarlo porque PAM ofrece sólo una solución y esto no quiere decir que sea la única, es muy probable que se generasen más óptimos. En otras palabras, en un algoritmo combinatorio como PAM, se produce más de un óptimo global. En particular, cuando en la implementación se comparan soluciones y estas son iguales, el programador escoge cualquiera de ellas porque todas las soluciones óptimas generadas son “igual de buenas”. Esto implica que una de ellas coincidirá con precisión con la solución exacta que resulta de la P-mediana codificada en Lingo y otras veces no serán iguales, pero esto no significa que no exista tal óptimo de la P-mediana en PAM.

Capítulo 5. P-Mediana y Particionamiento

5.1 La P-Mediana y el problema de Conglomerado o Particionamiento en la Investigación de Operaciones

Cuando un problema de Diseño Territorial (DT) es modelado bajo condiciones del Particionamiento clásico, tal problema suele llamarse Particionamiento Territorial (PT) debido no solo a la modelación, también a la solución computacional que utiliza técnicas algorítmicas de clasificación por particiones. En este caso, el Particionamiento puede verse como una metodología algorítmica considerando que el problema implícito de DT debe satisfacer criterios de compacidad y conexidad, cuya notación aquí es CC_DT.

En la literatura se pueden hallar modelos de Particionamiento de conjuntos muy parecidos entre sí, los cuales consiguen ajustarse al CC_DT ¹⁸. Específicamente, el Particionamiento para el caso de problemas territoriales consiste en agrupar pequeñas áreas geográficas llamadas unidades básicas en un número dado de grupos más grandes denominados territorios. Esta descripción requiere de un modelo para su solución, siendo la definición de partición discreta la que se adecúa para propósitos de este capítulo.

Obsérvese que la definición de diseño de zona (DZ) y partición comparten propiedades en su definición, es decir, las restricciones del problema de diseño de zonas son similares a las características del problema de clustering ¹⁹.

¹⁸ Nótese que solo se mencionan a los modelos de Particionamiento y se excluyen los modelos de clasificación jerárquica, no porque su utilidad sea irrelevante, sino porque este apartado se dedica al modelo de Particionamiento en DT. Por otra parte, clustering, agrupamiento, y conglomerado se refieren a la acción de agrupar en un sentido amplio y se toman como sinónimos cuando no tienen “apellido”.

¹⁹ Diseño de zonas, diseño territorial, diseño de territorio en un sentido pragmático, comparten la misma semántica, incluso zonificación o regionalización (Kalcsics *et al.*, 2005) (Hess *et al.*, 1971) (Zoltners & Shina, 1983).

Definición. Sea el conjunto inicial de unidades de área $X = \{x_1, x_2, \dots, x_n\}$ donde x_i es la i th unidad de área. El número de zonas es denotado por k y Z_i es el conjunto de todas las unidades de área que pertenecen a la zona Z_i (Shirabe, 2005). Entonces:

$$Z_i \neq \emptyset \text{ para } i = 1 \dots k, \quad (1)$$

$$Z_i \cap Z_j = \emptyset \text{ para } i \neq j \text{ y} \quad (2)$$

$$\bigcup_{i=1}^k Z_i = X \quad (3)$$

Por tanto, (1), (2) y (3) constituyen el conjunto de “restricciones equivalentes” tanto en el clustering como en el diseño de zonas.

Se establece que estos criterios cumplen de forma indirecta al menos la compacidad geométrica, aunque en distintos problemas resueltos computacionalmente, se ha observado gráficamente que compacidad y conexidad se satisfacen. En otros casos, se ha identificado en mapas que la contigüidad también se cumple, pero esta observación visual es válida solo para algunas aplicaciones como lo es la P-mediana y no se garantiza que la contigüidad sea alcanzada, incluso algunos autores ya han advertido que la inclusión de contigüidad representa un problema más complejo (Macmillan, 1994).

Definición. Una partición de un conjunto A de n elementos en k pares, es una familia de k subconjuntos disjuntos no vacíos de A , tales que su unión es el propio conjunto A , de esta forma A_1, \dots, A_k , son los subconjuntos de la partición, entonces se cumple:

$$A_j \neq \emptyset \quad (1)$$

$$A_i \cap A_j = \emptyset \text{ si } i \neq j \text{ y} \quad (2)$$

$$A_1 \cup A_2 \cup \dots \cup A_k = A \quad (3)$$

5.2 Problema de Particionamiento territorial como un problema de la P-mediana

Una aplicación de DT que puede ser empleada como metodología cuando se tratan criterios de compacidad y conexidad, son los modelos de localización-asignación. Particularmente la P-mediana es útil en estas condiciones

independientemente del enfoque de investigación de operaciones (entero-binario) o del problema combinatorio (abordado con metaheurísticas) ^{20, 20.1}.

El problema de la P-mediana ha sido ampliamente estudiado en la literatura. En (Reese, 2006) se puede encontrar una excelente revisión de metodologías para resolver la P-Mediana de forma aproximada y exacta o a través de grafos ²¹.

Los problemas de Particionamiento Territorial tienen aplicaciones diversas como son la identificación de distritos políticos, distritos escolares, instalaciones de servicios sociales, territorios comerciales, localización-asignación, etc. (Kalcsics et al., 2005) (Hess & Samuels, 1971) (Zoltners & Shina, 1983). En muchos trabajos,

²⁰ Cuando se habla aquí de modelo, se refiere a la representación matemática del problema y se traduce a una metodología siempre que los puntos del modelo se respeten y se adapten a una serie de instrucciones. Esto conduce a la elección de técnicas adecuadas que respondan al modelo, de tal modo que se identifiquen algoritmos candidatos para su solución y consecuentemente, evaluarlas para dar respuesta al modelo. Sin embargo, alguna variante en el modelo que no cambie el significado, implica una metodología razonablemente distinta pero que lleva a la misma solución, se dice así que el modelo de la P-mediana básico es el mismo en toda la literatura, aunque la metodología y la propuesta de algoritmo que se plantean para su solución sea distinta pero buscando el mismo resultado.

^{20.1} Por ejemplo, cuando se utilizan distintas metaheurísticas para dar respuesta a la P-mediana en el modelo combinatorio, las soluciones no son exactamente las mismas dado que son aproximadas, pero siempre se busca que tales aproximaciones sean cercanas al óptimo, o en el mejor de los casos al valor exacto. Para valorar las implementaciones, las instancias de prueba de la P-mediana se encuentran en el sitio web OR-Library. (<http://people.brunel.ac.uk/~mastjjb/jeb/info.html>)

²¹ La p-mediana puede establecerse en términos de grafos como: Sea $G = (V, E)$ un grafo no dirigido donde V es el conjunto de n vértices y E es el conjunto de aristas con un peso asociado que puede ser la distancia entre los vértices $d_{ij} = d(v_i, v_j)$ para toda $i, j = 1, \dots, n$ de acuerdo a determinada métrica, entonces con las distancias se forma una matriz simétrica, entonces se debe encontrar $V_p \subseteq V$ tal que $|V_p| = p$, donde p puede ser o bien variable o fijo, y que la suma de las distancias más cortas de los vértices en $\{V - V_p\}$ a su vértice más cercano en V_p se reduce al mínimo.

se utilizan criterios geográficos de medidas de adecuación de las soluciones. Los criterios comúnmente utilizados son la compacidad, conexidad y la contigüidad. De acuerdo con Kalcsics, un territorio es geográficamente compacto si tiene forma aproximadamente redonda y no está distorsionado, pero no existe una definición rigurosa del concepto (Kalcsics *et al.*, 2005).

En muchos problemas de carácter territorial, se acostumbra a utilizar como medida de compacidad la suma de distancias entre las unidades básicas en R^2 y el centroide al que están asignadas. Con al menos esa propiedad y respetando la teoría de la P-mediana, se modela el problema como la P-mediana y se asume que se satisface indirectamente conexidad y compacidad geométrica e incluso contigüidad, situación que debe formularse y demostrarse de acuerdo al caso específico.

5.3 Análisis del problema de conglomerado como uno de optimización

Sea el conglomerado un problema de agrupamiento por particiones, entonces dado un conjunto de n objetos denotado por $X = \{x_1, x_2, \dots, x_n\}$ en que $x_i \in R^D$ sea k un número entero positivo conocido apriori, el problema de clustering consiste en encontrar una partición:

$P = \{C_1, C_2, \dots, C_k\}$ de X , siendo C_j un conglomerado (grupo) conformado por objetos similares, satisfaciendo una función objetivo $f: R^D \rightarrow R$, y las condiciones: $C_i \cap C_j = \emptyset$ para $i \neq j$ y $\bigcup_{j=1}^k C_j = X$

Para medir la similaridad entre dos objetos x_a y x_b se usa una función de distancia denotada por $d(x_a, x_b)$, siendo la distancia euclidiana la más popular para medir la similaridad. Así la distancia entre dos diferentes elementos

$$x_i = (x_{i1}, \dots, x_{iD}) \quad y \quad x_j = (x_{j1}, \dots, x_{jD})$$

$$\text{es } d(x_i, x_j) = \sqrt{\sum_{l=1}^D (x_{il} - x_{jl})^2}$$

Los objetos de un conglomerado son similares cuando las distancias entre ellos es mínima; esto permite formular la función objetivo f , como

$$\sum_{j=1}^k \sum_{x_i \in C_j} d(x_i, \bar{x}_j) \quad (1)$$

Es decir, se desea minimizar (1); donde \bar{x}_j , conocido como elemento representativo del conglomerado (grupo), es la media de los elementos del conglomerado C_j , $\bar{x}_j = \frac{1}{|C_j|} \sum x_i \in C_j$ (2)

Y corresponde al centro de conglomerado. Bajo esas características, el clustering es un problema de optimización combinatoria, y ha sido demostrado que es NP-difícil.

5.4 Enfoque clásico P-mediana

La P-mediana consiste en que dado un conjunto de puntos (o ubicación de consumidores) y una matriz de distancias (o costos) entre todos y cada uno de los puntos, se eligen p puntos (o ubicación de las instalaciones) con el propósito de minimizar la suma de las distancias de todos los puntos al “punto elegido más cercano”. En 1970 ReVelle and Swain presentaron la primera formulación de programación entera para el p -median problema citado por Church (2003) (Church, 2008) (Drezner, 1995) (Daskin, 1995) (ReVelle & Swain, 1970). En general el problema de la p -mediana se puede ser expresado matemáticamente como un problema de optimización discreta.

Primero se denota a la matriz de distancias como d_{ij} , que expresa la distancia entre los puntos potenciales de ubicación i y los puntos de demanda j . La variable binaria x_{ij} corresponde a la asignación o no asignación de los puntos de demanda j a la instalación i . La variable binaria y_i indica que una instalación se establece en el punto i o no. El planteamiento como un problema entero binario es de la forma siguiente:

$$\begin{aligned} \text{Sean} \quad & x_{ij} = \begin{cases} 1 & \text{Si el punto } j \text{ se asigna al punto } i \\ 0 & \text{en otro caso;} \end{cases} \\ \text{y} \quad & y_i = \begin{cases} 1 & \text{Si en el punto } i \text{ se ubica una instalación} \\ 0 & \text{en otro caso;} \end{cases} \\ \min \quad & Z = \sum_{i=1}^k \sum_{j=1}^n d_{ij} x_{ij} \quad (a) \end{aligned}$$

$$\text{sujeto a } \sum_{i=1}^k x_{ij} = 1 \quad \forall j = 1, 2, \dots, n; \quad (b)$$

$$\sum_{j=1}^n x_{ij} \leq ny_i \quad \forall i = 1, 2, \dots, k; \quad (c)$$

$$\sum_{i=1}^k y_i = p \quad (d)$$

Donde k es el número de vértices potenciales donde se puede localizar una mediana generalmente $k = n$, p es el número fijo de medianas requeridas. La ecuación (a) es la función objetivo que minimiza la distancia del sistema, la restricción (b) establece que cada punto de demanda sólo puede ser asignado a una instalación, la restricción (c) establece que la asignación de puntos de demanda a cada una de las instalaciones o medianas y finalmente la restricción (d) garantiza que de entre los k puntos potenciales de ubicación se eligen exactamente p .

$$\text{Prueba. Definimos, } u_{jr} = \begin{cases} 1 & \text{si } r \text{ es asignado a } j \\ 0 & \text{en otro caso} \end{cases}$$

Como cada punto pertenece a un único grupo, se satisface $\sum_{j=1}^k u_{jr} = 1$

$$\text{Además, } \sum_{x_i \in C_j} d(x_i, \bar{x}_j) = \sum_{r=1}^n u_{jr} d(x_i, \bar{x}_j)$$

$$\text{Entonces } \sum_{j=1}^k \sum_{x_i \in C_j} d(x_i, \bar{x}_j) = \sum_{j=1}^k \sum_{r=1}^n u_{jr} d(x_i, \bar{x}_j)$$

Si el problema lo interpretamos como puntos potenciales de ubicación j y los puntos de demanda r . La variable binaria u_{jr} corresponde a la asignación de los puntos de demanda r a la instalación j o no. La variable \bar{x}_j se puede interpretar como una instalación se establece en el grupo C_j .

$$y_j = \begin{cases} 1 & \text{en la instalación } \bar{x}_j \\ 0 & \text{en otro caso} \end{cases}$$

$$\text{Entonces se satisface } \sum_{j=1}^k y_j = k \quad \text{y} \quad \sum_{r=1}^n u_{jr} = ny_j$$

En conclusión, se cumplen todas las hipótesis del modelo de la P-mediana y viceversa.

5.5 Transformación de Modelo del problema de la P-mediana de programación entera binaria a optimización combinatoria

Es de interés mostrar la transformación del problema de la P-mediana del modelo de programación entera binaria a uno de optimización combinatoria.

Para implementar una clase de algoritmos de aproximación en la búsqueda de soluciones a los problemas NP-duros, se debe plantear como un modelo de optimización combinatoria y el problema de la P-mediana es NP-duro (Kariv & Hakimi, 1979).

El problema de la P-mediana es el correcto en la solución de problemas de localización, de diseño territorial, particionalmento de conjuntos, de diseño de clúster, entre otros y es necesario implementar métodos de búsqueda de soluciones como las metaheurísticas (Daskin, 1995).

A las aplicaciones de los problemas de localización les corresponde resolver la ubicación de una o varias instalaciones, de tal manera que se optimice uno o varios objetivos, tal es el caso de costo de transporte, servicio a los clientes, partición del mercado, etc. El estudio de los problemas de localización involucra muchos campos del conocimiento como investigación de operaciones, ciencias de la administración, ingeniería industrial, ciencias de la computación, planeación urbana y distintas áreas relacionadas. Entre las aplicaciones al diseño de redes logísticas se encuentran el diseño territorial, ubicación de almacenes, plantas de producción, plantas de ensamble, hospitales, estaciones de bomberos, estaciones de policía, escuelas, etc.

Para mostrar la transformación del modelo matemático del problema de la P-mediana como uno de programación entera binaria a uno optimización combinatoria, se tiene la siguiente nomenclatura:

PPM= problema de la p-mediana

PPEB=problema de programación entera binaria

POC= problema de optimización combinatoria

Primero es importante una introducción al modelo de optimización combinatoria de la P-mediana. Sea el problema de la P-Mediana (PPM) de la siguiente manera:

Dado un conjunto de n vértices de un grafo en el plano denotado por $V = \{1, 2, \dots, n\}$, $|V| = n$, el objetivo es encontrar un subconjunto de vértices $L \subseteq V$, con $|L| = p$ de potenciales ubicaciones de las medianas tal que la distancia media total del diseño sea mínima. Decimos que este planteamiento es de tipo Problema de Optimización Combinatoria (POC) ya que el espacio de factibilidad Ω del PPM son todos los subconjuntos L de cardinalidad p de un conjunto de cardinalidad n , con $p < n$, esto es:

$$|\Omega| = \frac{n!}{p!(n-p)!} = \binom{n}{p}$$

En general, una instancia para el PPM es denotada por $\text{PPM}(n, D, p)$, donde n es la cardinalidad del conjunto V , p es la cardinalidad de los subconjuntos obtenidos y $D = (d_{ij})$ es la matriz de distancias entre todos los pares de vértices de V .

Segundo, sea el problema de la P-mediana visto como un modelo entero-binario, entonces el modelo matemático del PPEB del PPM se desarrolla como sigue:

El PPM sobre un grafo se puede expresar matemáticamente como un problema de programación entero binario (PPEB) de la forma: (Church, 2003).

$$\text{Sea } x_{ij} = \begin{cases} 1 & \text{Si el vertice } j \text{ se asigna al vertice } i \\ 0 & \text{en cualquier otro caso} \end{cases}$$

$$\text{Sea } y_i = \begin{cases} 1 & \text{si el vertice } i \text{ es una mediana} \\ 0 & \text{en cualquier otro caso} \end{cases}$$

$$\text{Min } Z = \sum_{i=1}^k \sum_{j=1}^n d_{ij} x_{ij} \quad (1)$$

$$\text{sujeto a } \sum_{i=1}^k x_{ij} = 1 \quad \forall j = 1, \dots, n \quad (2)$$

$$\sum_{j=1}^n x_{ij} \leq n y_i \quad \forall i = 1, \dots, k \quad (3)$$

$$\sum_{i=1}^k y_i = p \quad (4)$$

La ecuación (1) es la función objetivo con variables de decisión x_{ij} y y_i binarias. El número de vértices potenciales donde se puede localizar una mediana es k y generalmente $k = n$. El número fijo de medianas requeridas es p . Las restricciones (2) garantizan que todo vértice tiene asociada una mediana. Las restricciones (3) indican la distribución de los vértices a las medianas. Las (4) determina el número de medianas.

5.5.1 El procedimiento para la transformación a través del análisis de un caso

El PPM como problema de programación entera binaria PPEB tiene un espacio de factibilidad del tipo exponencial 2^n y como un POC es $\binom{n}{p}$

Entonces se define una transformación $T: 2^n \rightarrow \binom{n}{p}$

Consideremos un caso de un grafo de $n = 17$ vértices, es decir, $V = \{1, 2, \dots, 17\}$ ilustrado en la figura 6, cuya solución se determinó por el modelo PPEB que se observa en la figura 7. Los resultados con un costo C determinados por la función objetivo de la ecuación (1), se representan en las siguientes figuras:

Figura 6

Grafo de 17 vértices

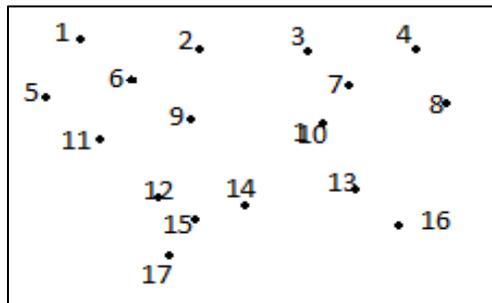


Figura 7

Caso de $n = 17$ con $p = 4$

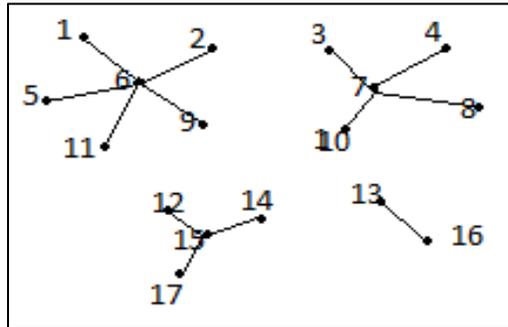


Tabla 6

Soluciones binarias

Solución binaria	Mediana	Grupo
$x_{61} = 1, x_{62} = 1, x_{65} = 1, x_{69} = 1, x_{611} = 1$	6	6, 1, 2, 5, 9, 11
$x_{73} = 1, x_{74} = 1, x_{78} = 1, x_{710} = 1$	7	7, 3, 4, 8, 10
$x_{1512} = 1, x_{1514} = 1, x_{1517} = 1$	15	15, 12, 14, 17
$x_{1316} = 1$	13	13, 16

La solución se muestra en la tabla 6 y la solución binaria determina las medianas en $y_6 = 1, y_7 = 1, y_{15} = 1, y_{13} = 1$.

La consecuente solución combinatoria conseguida con un algoritmo enumerativo de Particionamiento, son los “centroides” $L = \{6, 7, 13, 15\}$. Dicha solución tiene un costo C obtenido de la suma de las sumas de las distancias de las “medianas”, ahora expresadas como centroides a sus vértices asociados, es decir

$$C = d(6,1) + d(6,2) + d(6,5) + d(6,9) + d(6,11) + d(7,3) + d(7,4) + d(7,8) + d(7,10) + d(15,12) + d(15,14) + d(15,17) + d(13,16).$$

La instancia de prueba para el problema combinatorio de la P-Mediana $(17, D_{17 \times 17}, 4)$ ha mostrado la equivalencia entre las soluciones del PEB y el POC para el PPM.

Desafío: Generalizar la transformación.

Referencias

- Aldenderfer, M. S. & Blashfield, R K. (1984). *Cluster Analysis. Series: Quantitative Applications in the Social Sciences*. California Sage Publications, Inc.
- Aloise, D. (2009). *Exact algorithms for minimum sum-of-squares clustering*. Ecole Polytechnique Montreal (ACM Digital Library).
- Amoroso, M. S. & Ávila, H. S. (2015). *Aplicación de las técnicas de agrupamiento para la distribución cuasi-óptima de una red híbrida WDM-TDM/PON en cascada multinivel que da soporte a una Smart Grid o Smart City* [tesis de Ingeniero Electrónico, Universidad Politécnica Salesiana]. Repositorio Institucional. <http://dspace.ups.edu.ec/handle/123456789/7736>.
- Anil, J. (2012). *Data Clustering: 50 Years Beyond K-Means*. Michigan State University.
- Ankerst, M., Breunig, M. M., Kriegel, H. P. & Sander, J. (1999, Jun 3). *OPTICS: Ordering points to identify the clustering structure* [conference session]. *ACM SIGMOD Record*. <https://dl.acm.org/doi/10.1145/304182.304187>
- Atkeson, C., Moore, A. & Schaal, S. (1997). Locally Weighted Learning. *Artificial Intelligence Review*, 11, 11-73. <https://doi.org/10.1023/A:1006559212014>
- Ayala, J., Aguilar, I., García, F. & Gómez, H. (2019) *Introducción al análisis de algoritmos* (1ª ed.). Editorial Centro de Estudios e Investigaciones para el Desarrollo Docente.
- Baeza, R. & Ribeiro, B. (1996). *Modern information retrieval: The concepts and technology behind search*. Reading M.A., Addison-Wesley.
- Bailey, K. (1994). *Typologies and Taxonomies: An Introduction to Classification Techniques* (vol 102). Quantitative Applications. SAGE Publications.
- Bakamidis, S. G. (1993, April 30). *An exact fast nearest neighbour identification technique* [conference session]. IEEE International Conference on Acoustics, Speech and Signal processing, Minneapolis, MN, USA. <https://doi.org/10.1109/ICASSP.1993.319898>
- Bernábe, B. (2010). *Diseño y desarrollo de un modelo para la determinación de zonificación óptima* [tesis de doctorado, Universidad Nacional Autónoma de México]. Repositorio de Tesis DGBSDI. https://ru.dgb.unam.mx/handle/DGB_UNAM/TES01000652934
- Bernábe-Loranca, M. B., González-Velazquez, R., Guillén Galván, C & Granillo-Martínez, E. (2021, December 15). *A model of compactness-homogeneity for territorial design* [conference session]. 21st International Conference on

- Bichot, C. E. & Siarry, P. (2013). *Graph Partitioning*. ISTE. Wiley.
- Bell, D. (1958). The principles of sorting. *The Computer Journal*, 1(2), 71–77.
<https://doi.org/10.1093/comjnl/1.2.71>
- Bose, R. C. & Nelson, R. J. (1962). A sorting problem. *Journal of the ACM (JACM)*, 9(2), 282–296. <https://doi.org/10.1145/321119.321126>
- Carrera, J. V. (2005). *Clasificación temática de textos basada en transformaciones semánticas* [tesis de doctorado, Instituto Politécnico Nacional].
<https://www.saber.cic.ipn.mx/SABERv3/Repositorios/webVerArchivo/25861/3>
- Cattell, R. (1943). *The Description of Personality: Basic Traits Resolved Into Clusters*. American psychological association.
- Curry, H. B. (2010) *Foundations of Mathematical Logic (Dover Books on Mathematics)* (2^a ed). Dover Publications.
- Church, R. L. (2003). COBRA: A New Formulation of the Classic p-Median Location Problem. *Annals of Operations Research*, 122, 103–120.
<https://doi.org/10.1023/A:1026142406234>
- Church, R. L. (2008). BEAMR: an exact and approximate model for the p-median problema. *Computers and Operation Research*, 35(2), 417-26.
<https://doi.org/10.1016/j.cor.2006.03.006>
- Daskin, M. S. (1995). *Network and Discrete Location: Models, Algorithms, and Applications*. John Willey&Sons, Inc.
- Deerwester, S., Dumals, S., Furnas, T., Landauer, G. & Harshman, R. (1990). Indexing by latent semantic analysis. *Journal of the American Society for Information Science*, 41(6), 391-407. [https://doi.org/10.1002/\(SICI\)1097-4571\(199009\)41:6<391:AID-ASI1>3.0.CO;2-9](https://doi.org/10.1002/(SICI)1097-4571(199009)41:6<391:AID-ASI1>3.0.CO;2-9)
- Delattre, M. & Hansen, P. (1980). Bicriterion cluster analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-2(4), 277-291.
<https://doi.org/10.1109/TPAMI.1980.4767027>
- Díaz, G. J., Bernábe, L. B., Olivares, B. E., Luna R. D., Martínez, F. J., (2012). Relajación lagrangeana para el problema de Particionamiento de áreas geográficas. *Revista de Matemática: Teoría y Aplicaciones*, 19(2),169-181.
<https://www.redalyc.org/articulo.oa?id=45326926004>

- Diday, E. (1972). Optimisation en classification automatique et reconnaissance des forms. *R.A.I.R.O. Recherche opérationnelle*, 6(V3), 61-95. <https://doi.org/10.1051/ro/197206V300611>
- Diday E., Lemaire J., Pouget J. & Testu F. (1982). *Eléments d'Analyse de Données*. Dunod.
- Drezner, Z. (1995). *Facility location: A survey of applications and methods*. Springer.
- Dohan, D., Karp, S. & Matejek., B. (2015). *K-median algorithms: theory in practice*. <https://www.cs.princeton.edu/courses/archive/fall14/cos521/projects/kmedian.pdf>
- Egner, R. E., Denonn, L.E. (1961). Bertrand Russell Escritos básicos 1903-1959, Allen & Unwin.
- Estivill-Castro, V. (2002). Why so many clustering algorithms — A Position Paper. *ACM SIGKDD Explorations Newsletter*, 4(1), 65-75. doi:10.1145/568574.568575.
- Everitt, B. S. (1993). *Cluster Analysis* (3^a ed.). Edward Arnold.
- Fraenkel, A. A., Bar-Hillel, Y., Lévy, A. (1973). *Foundations of Set Theory* (2^a ed, Vol. 67), Elsevier Science.
- Ferreira, C. E., Martin, A., de Souza, C.C., Weismantel, R. & Wolsey, L. A. (1998). The node capacitated graph partitioning problem: A computational study. *Mathematical Programming*, 81, 229-256. <https://doi.org/10.1007/BF01581107>
- Firdaus, S. & Uddin, M. A. (2015). A survey on clustering algorithms and complexity analysis. *IJCSI International Journal of Computer Science Issues*, 12(2), 62-85. <https://www.ijcsi.org/papers/IJCSI-12-2-62-85.pdf>
- Forgy, E. W. (1965). *Cluster analysis of multivariate data: efficiency versus interpretability of classifications*. *Biometrics*, 21, 768-780 <https://www.semanticscholar.org/paper/Cluster-analysis-of-multivariate-data-%3A-efficiency-Forgy/5c4feeae0d911e30866b7149c1195cd8c007199b>
- Friedman, J., Hastie, T. & Tibshirani, R. (2001). *The elements of statistical learning* (Vol. 1). Springer.
- García, F. (2001). *Matemática discreta*. Thomson Learning.
- Gersho, A. & Gray, R. (1991). *Vector Quantization and Signal Compression*. Kluwer Academic.

- Gierz, G., Hofmann, K. H., Keimel, K., Lawson, J. D., Mislove & Scott, D. S. (2003). *Continuous Lattices and Domains*. Encyclopedia of Mathematics and its Applications, Vol. 93, Cambridge University Press
- Giraldo, F. A., León, E. & Gómez, J. (2013). Caracterización de flujos de datos usando algoritmos de agrupamiento. *Tecnura*, 17(37), 153-166. <https://doi.org/10.14483/udistrital.jour.tecnura.2013.3.a13>
- Gordon, A. D. (1987). A review of hierarchical classification. *Journal of the Royal Statistical Society, Series A (General)*, 150(2), 119–137. <http://dx.doi.org/10.2307/2981629>
- Han, J., Kamber, M. & Tung, A. K. H. (2001). Spatial Clustering Methods in Data Mining: A Survey. In H. J. Miller & J. Han (eds.), *Geographic Data Mining and Knowledge Discovery, Research Monographs in GIS* (pp 187 217). Taylor and Francis
- Han, J. & Kamber, M. (2006). *Data mining: Concepts and techniques*, (7^a ed.). Morgan Kaufmann.
- Hansen, P. & Delattre, M. (1978). Complete-link cluster analysis by graph coloring. *Journal of the American Statistical Association*, 73 (362), 397-403. <https://doi.org/10.2307/2286672>
- Hansen, P., Jaumard, B. & Frank, O. (1989). Maximum sum-of-splits clustering. *Journal of Classification*, 6, 177-193. <https://doi.org/10.1007/BF01908598>
- Hansen, P. & Aloise, D. (2009). A survey on exact methods for minimum sum-of-squares clustering, pp 1-2, <http://www.math.iit.edu/Buck65files/msscStLouis.pdf>
- Hansen, P. & Jaumard, B. (1997). Cluster analysis and mathematical programming. *Mathematical Programming*, 79, 191-215. <https://doi.org/10.1007/BF02614317>
- Hartigan, J. (1975). *Cluster algorithms*. John Wiley & Sons.
- Hartigan, J. A. (1974). Clustering Algorithms. John Wiley & Sons, Nueva York. Pp. 251-276 *Math. Sci. Hum*, 82 (1983), 85-92. [https://people.inf.elte.hu/fekete/algorithmusok_msc/klaszterezes/John%20A.%20Hartigan-Clustering%20Algorithms-John%20Wiley%20&%20Sons%20\(1975\).pdf](https://people.inf.elte.hu/fekete/algorithmusok_msc/klaszterezes/John%20A.%20Hartigan-Clustering%20Algorithms-John%20Wiley%20&%20Sons%20(1975).pdf)
- Hastie, T., Tibshirani, R., & Friedman, J. (2009). *The Elements of Statistical learning: Data Mining, Inference, and Prediction*. (2^a ed). Springer.
- Hernández, S. (2009). *Clasificadores rápidos basados en el algoritmo del vecino más similar para datos mezclados* [tesis de doctorado, Instituto Nacional de

Astrofísica, Óptica y Electrónica]. INAOE Repositorio.
<http://inaoe.repositorioinstitucional.mx/jspui/handle/1009/386>

Hess, S. W. & Samuels, S. A. (1971). Experiences with a sales districting model: criteria and implementation. *Management Science*, 18(4), 41-54.
<https://doi.org/10.1287/mnsc.18.4.P41>

Hinneburg, A. & Keim, D. A. (1998). An Efficient Approach to Clustering in Large Multimedia Databases with Noise. Institute of Computer Science, University of Halle, Germany, American Association for Artificial Intelligence KDD-98, pp. 58-65.

Holmes, M. R. (2012). Alternative Axiomatic Set Theories. In Edward N. Zalta (Eds.), *Handbook of the History of Logic The Stanford Encyclopedia of Philosophy*. <https://plato.stanford.edu/archives/win2021/entries/settheory-alternative>

Huertas, A. & Manzano, M. (2002). *Teoría de Conjuntos*.
<https://webs.ucm.es/info/pslogica/teoriaconjuntos.pdf>

IEEE Standard for Floating-Point Arithmetic. (2019). IEEE Std 754-2019 (Revisión de IEEE, 754(2008)), 1-84. doi: 10.1109/IEEESTD.2019.8766229

Jain, A. K. (2010). Data Clustering: 50 Years Beyond K-Means. *Pattern Recognition Letters*, 31(8), 651-666.
<https://doi.org/10.1016/j.patrec.2009.09.011>

Johnson, S. C. (1967). Hierarchical Clustering Schemes. *Psychometrika*, 32, 241-254. <https://doi.org/10.1007/BF02289588>

Kalcsics, J., Nickel, S. & Schröder, M. (2005). Toward a unified territorial design approach: Applications, algorithms, and GIS integration, *TOP*, 13(1), 1–56.
<https://doi.org/10.1007/BF02578982>

Kanungo, T., Netanyahu, N. & Wu, A. (2002). An efficient K means algorithm: Analysis and Implementation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(7), 881–892.
<https://www.cs.umd.edu/~mount/Projects/KMeans/pami02.pdf>

Kaufman, L. & Rousseeuw, P. (1987). *Clustering by means of medoids*. In Y. Dodge (Ed.), *Statistical Data Analysis, based on the L1 Norm* (pp. 405-416), Elsevier/North Holland, Amsterdam.

Kaufman, L. & Rousseeuw, P. J. (2005). *Finding Groups in Data: An Introduction to Cluster Analysis*. John Wiley, Hoboken,
<http://dx.doi.org/10.1002/9780470316801.ch1>

- Kariv, O., & Hakimi, S. L. (1979). An Algorithmic Approach to Network Location Problems. Part II: The p-Medians. *SIAM Journal on Applied Mathematics*, 37(3), 539–560. <http://www.jstor.org/stable/2100911>
- Knuth, D. (1998). *The Art of Computer Programming: Sorting and Searching*. (2^a ed). Addison Wesley.
- Lipschutz, S. (1970). *Teoría de Conjuntos y Temas Afines, Teoría y 530 problemas resueltos, Serie de compendios SCHAUM*. Mc Graw-Hill.
- Macmillan, W. (1994). Optimization modelling in GIS framework: the problem of political redistricting. In Fotheringham and P. Rogerson (Eds.), *Spatial Analysis and GIS* (pp 221-246). Taylor & Francis.
- MacQueen, J. (1966, 7 January). *Some methods for classification and analysis of multivariate observations* [conference sesión]. 5th Symposium of Mathematics, Statistics and Probability, University of California Press, Berkeley. https://books.google.com.gt/books?id=IC4Ku_7dBFUC
- Mark, J. V., Katherine, S. P. & Jennifer, B. (2002). A New Partitioning Around Medoids Algorithm. *Journal of Statistical Computation and Simulation*, 73(8), 575-584. <https://www.tandfonline.com/doi/abs/10.1080/0094965031000136012>
- Martin, W. A. (1971). Sorting. *ACM Computing Surveys*, 3(4), 147–174. <https://dl.acm.org/doi/10.1145/356593.356594>
- Mayer-Schönberger, V. & Cukier, K. (2013). *Big Data: A Revolution That Will Transform How We Live, Work and Think*. Publisher John Murray.
- Merle, D., Hansen, P., Jaumard, B & Mladenovic, N. (2000). An interior point algorithm for minimum sum-of-squares clustering. *SIAM J. SCI. Comput. Society for Industrial and Applied Mathematics*, 21(4), 1485–1505. <https://epubs.siam.org/doi/abs/10.1137/S1064827597328327?mobileUi=0>
- Minkowski, H. (1953). *Geometrie der Zahlen*. Chelsea.
- Müllner, D. (2013). Fastcluster: Fast Hierarchical, Agglomerative Clustering Routines for R and Python. *Journal of Statistical Software*, 53(9), 1–18. <https://doi.org/10.18637/jss.v053.i09>
- Muñoz, J. M. (2002). *Introducción a la Teoría de Conjuntos* (4^a ed.). Universidad Nacional de Colombia.
- Nijijima, S., & Kuhara, S. (2005, December 19). *Effective nearest neighbor methods for multiclass cancer classification using microarray data* [conference session]. 16th International Conference on Genome Informatics, Pacifico Yokohama, Japan. <http://giw.hgc.jp/giw2005/program.pdf>

https://www.researchgate.net/publication/239947005_A_Classification_Framework_Applied_to_Cancer_Gene_Expression_Profiles

- Newmann, J. V. (1925). Eine Axiomatisierung der Mengenlehre. *Journal für die reine und angewandte Mathematik*, 154, 219–240. https://gdz.sub.uni-goettingen.de/id/PPN243919689_0154
- Oubiña, L. (1974). *Introducción a la Teoría de Conjuntos* (7ª ed.). Eudeba.
- Park, H. S. & Jun, C. H. (2009). A simple and fast algorithm for K-medoids clustering. *Expert Systems with Applications*, 36(2), 3336-3341. <https://dl.acm.org/doi/10.1016/j.eswa.2008.01.039>
- Pérez, M. A. (2007). *Mejora al algoritmo de agrupamiento K-means mediante un nuevo criterio de convergencia y su aplicación a bases de datos poblacionales de cáncer*. II Taller Latino Iberoamericano de Investigación de Operaciones.
- Pinzón, A. (1975). *Conjuntos y Estructuras*. Harla.
- Prisco, C. A. D. (1997). *Una introducción a la teoría de conjuntos y los fundamentos de la matemática*. Centro de Lógica, Epistemología e História da Ciência, Universidade Estadual de Campinas, S. P.
- Piza, E., Murillo, A. & Trejos, J. (1999). Nuevas técnicas de Particionamiento en clasificación automática. *Revista de Matemática: Teoría y Aplicaciones*, 6(1), 51–66. <https://revistas.ucr.ac.cr/index.php/matematica/article/view/168>
- Rattigan, M. J., Maier, M. & Jensen, D. (2007, June 20). *Graph clustering with network structure indice* [conference session]. 24th international conference on Machine learning, Corvalis Oregon. <https://dl.acm.org/doi/10.1145/1273496.1273595>
- Reese, J. (2006). Solution methods for the p-median problem: An annotated bibliography, *Networks* 48(3): 125–142. <https://onlinelibrary.wiley.com/doi/10.1002/net.20128>
- Régnier, S. (1983). Sur quelques aspects mathématiques des problèmes de classification automatique, *Mathématiques et sciences humaines*, 82, 13-29. http://www.numdam.org/article/MSH_1983__82__13_0.pdf
- ReVelle, C. S. and Swain, R. W. (1970). "Central Facilities Location," *Geographical Analysis*, 2(1), 30-42. <https://onlinelibrary.wiley.com/doi/10.1111/j.1538-4632.1970.tb00142.x>
- Rincón, B. (2010). Diseño de zonas geoméricamente compactas utilizando celdas cuadradas [tesis de doctorado, Universidad Nacional Autónoma de México].

- Robert, G., & Ross, I. (1993). *The R Project for Statistical Computing*. (version 3.5.1) [software]. R. <https://www.r-project.org/>
- Roberts, S. J. (1997). Parametric and non-parametric unsupervised cluster analysis. *Pattern Recognition*, 30(2), 261-272. [https://www.scirp.org/\(S\(351jmbntvnsjt1aadkposzje\)\)/reference/ReferencesPapers.aspx?ReferenceID=1884877](https://www.scirp.org/(S(351jmbntvnsjt1aadkposzje))/reference/ReferencesPapers.aspx?ReferenceID=1884877)
- Romero, D., Burguete, J., Martínez, L. E. & Velasco, J. R. (2006). *Un enfoque de optimización combinatoria para la construcción de marcos de muestreo en hogares*. Boletín de los Sistemas Nacionales Estadístico y de Información Geográfica, Instituto Nacional de Estadística, Geografía e Informática, 2(1), 44-51. https://www.inegi.org.mx/contenido/productos/prod_serv/contenidos/espanol/bvinegi/productos/integracion/especiales/BoletinSNEIG/2006/Bsneig3.pdf
- Romero, M., González, R., Estrada, M., Martínez, J. L. & Bernábe, M. B. (2019). Solution Search for the Capacitated P-Median Problem using Tabu Search. *International Journal of Combinatorial Optimization Problems and Informatics*, 10(2), 17-25. <https://www.ijcopi.org/ojs/article/view/118>
- Sebastiani, F. (2002). Machine learning in automated text categorization. *ACM Computing Surveys*, 34(1), 1-47. <https://dl.acm.org/doi/10.1145/505282.505283>
- Shirabe, T. (2005). *A Model of Contiguity for Spatial Unit Allocation*. *Geographical Analysis*, 37(1), 2-16. <https://onlinelibrary.wiley.com/doi/abs/10.1111/j.1538-4632.2005.00605.x>
- Skiena, S. (1990). *Implementing Discrete Mathematics: Combinatorics and Graph Theory with Mathematica*, Addison-Wesley.
- Sloane, N. J. A. (2022). The on-line encyclopedia of integer sequences. OEIS Foundation Inc. <https://oeis.org/A000110>
- Steven, R. (2008). *Lattices and Ordered Sets*. Springer.
- Struyf, A., Hubert, M., & Rousseeuw, P. (1997). Clustering in an Object-Oriented Environment. *Journal of Statistical Software*, 1(4), 1–30. <https://doi.org/10.18637/jss.v001.i04>
- Tan, P., Steinbach, M. & Kumar, V. (2005). *Introduction to Data Mining*. Addison-Wesley Longman Publishing Co., Inc.

- Trejos, Z. J. (2009). *Modelos de Clasificación Automática, Centro de Investigación en Matemática Pura y Aplicada (CIMPA)*. Universidad de Costa Rica.
- Tryon, R. (1939). *Cluster Analysis: Correlation Profile and Orthometric (factor) Analysis for the Isolation of Unities in Mind and Personality*. Edwards brother, Incorporated, lithoprinters and publishers.
- Tsybakov, A. B. (2009). *Introduction to nonparametric estimation*. Springer Science+Business Media.
- Varela, F. (2018). Un estudio computacional de diferentes técnicas de clustering. [tesis de maestría en técnicas estadísticas, Universidad de Santiago de Compostela]. Repositorio de Tesis http://eio.usc.es/pub/mte/descargas/ProyectosFinMaster/Proyecto_1354.pdf
- Varela, F. (2018). Un estudio computacional de diferentes técnicas de clustering. [tesis de maestría en técnicas estadísticas, Universidad de Santiago de Compostela]. Repositorio de Tesis http://eio.usc.es/pub/mte/descargas/ProyectosFinMaster/Proyecto_1354.pdf
- Von-Luxburg, U., Williamson, R. C. & Guyon, I. (2011, July 2). *Clustering: Science or Art?* [conference session]. JMLR: Workshop on Unsupervised and Transfer Learning, Bellevue, Washington, USA <http://proceedings.mlr.press/v27/luxburg12a/luxburg12a.pdf>
- Vicente, E., Rivera, L., & Mauricio, D. (2005). GRASP en la resolución del problema de clustering. *Revista de Investigación de Sistemas e Informática*, 2(2), 16–25. <https://revistasinvestigacion.unmsm.edu.pe/index.php/sistem/article/view/3110>
- Villagra, A., Guzman, A., Pandolfi, D. & Leguizamon, G. (2009). Análisis de medidas nosupervisadas de calidad en clusters obtenidos por K-means y Particle Swarm Optimization. *Revista Ciencia y Tecnología, edición especial CICA 2009*, 1(1), 1-8. <https://dspace.palermo.edu/ojs/index.php/cyt/article/view/782/684>
- Xu, R. & Wunsch, D. (2005). Survey of clustering algorithms. *IEEE Transactions on neural networks*, 16(3), 645-678. <https://ieeexplore.ieee.org/document/1427769>
- Ward, J. H. (1963). Hierarchical Groupings to optimise an objective function. *Journal of the American Statistical Association*, 58(301), 236-244. <https://www.tandfonline.com/doi/abs/10.1080/01621459.1963.10500845>
- Webb, A. R. (2003). *Statistical pattern recognition*. John Wiley & Sons.
- Wolf, G. (2012). Foundations of location analysis. *International Series in Operations Research & Management Science*, 26(3), 577-578.

<https://www.tandfonline.com/doi/abs/10.1080/13658816.2011.631498?journalCode=tgis20>

Yolis, E. (2003). Algoritmos genéticos aplicados a la categorización automática de documentos, *Revista Eletrônica de Sistemas de Informação*, 2(2). doi: 10.5329/RESI.

Zhang, Q., Couloigner, I. (2005, may 9). *A New and Efficient K-Medoid Algorithm for Spatial Clustering* [conference session], ICCSA Computational Science and Its Applications. https://doi.org/10.1007/11424857_20

Zoltners, A. & Sinha, P. (1983). Towards a unified territory alignment: A review and model. *Management Science*, 29(11) 1237-1256. <https://pubsonline.informs.org/doi/abs/10.1287/mnsc.29.11.1237>