Universidad del Bío-Bío. Sistema de Bibliotecas - Chile

Magíster en Matemática
Mención Matemática Aplicada o Mención Estadística

Método de agrupamiento α-Condorcet

Tesis presentada al Programa de Magíster en Matemática, Mención (Matemática Aplicada o Mención Estadística) del Departamento de Estadística y Departamento de Matemática de la Universidad del Bío Bío como parte de los requisitos para la obtención del grado de Magíster en Matemática con Mención en (Matemática Aplicada o Estadística) de la Universidad del Bío-Bío.

by José Avilez Bozo

Director: Tarik Faouzi Nadim

Co-director: Luis Firinguetti Limone

Universidad del Bío-Bío. Sistema de Bibliotecas - Chile

Universidad del Bío-Bío. Sistema de Bibliotecas - Chile

Método de agrupamiento α-Condorcet

por

José Avilez Bozo

Tesis presentada al Programa de Magíster en Matemática, Mención (Matemática Aplicada o Mención Estadística) del Departamento de Estadística y Departamento de Matemática de la Universidad del Bío Bío como parte de los requisitos para la obtención del grado de Magíster en Matemática con Mención en (Matemática Aplicada o Estadística) de la Universidad del Bío-Bío.

Aprobada por:

Dr. Tarik Faouzi Nadim
Director
Departamento de Estadística , Universidad del Bío-Bío

Dr. Luis Firinguetti Limone
Co-Director
Departamento de Estadística , Universidad del Bío-Bío

Dr. Christian Caamaño Carrillo
Profesor Informante
Departamento de Estadística , Universidad del Bío-Bío

Dr. Rubén Carvajal Schiaffino (externo)
Profesor Informante
Departamento de Matemática y Ciencia de la Computación (DMCC),
Universidad de Santiago de Chile

Universidad del Bío-Bío
Facultad de Ciencias
Departamento de (Matemática o Estadística)

Universidad del Bío-Bío. Sistema de Bibliotecas - Chile

RESUMEN

Establecer el número de grupos bajo los métodos de agrupamiento es fundamental para muchas áreas de estadística aplicada, ciencias sociales y aprendizaje automático, entre otros. Motivado por un problema que surge de cómo definir o asignar una carga factorial para cada variable latente, a partir del análisis factorial confirmatorio, consideramos un nuevo enfoque de clúster de Condorcet con el número de grupos fijo. En particular, definimos una nueva función de costo a minimizar para encontrar la asignación de clúster óptima, fijando el número de grupos. El método propuesto es un válido competidor del método K-means en términos de la solución óptima.

Palabras Claves:K-means, Condorcet clustering, datos categóricos, ciencias sociales

Universidad del Bío-Bío. Sistema de Bibliotecas - Chile

ABSTRACT

Setting the number of groups under clustering methods is fundamental to many areas of applied statistics, social science and machine learning, amongst others. Motivated by a problem coming from how to define or assign a factor loading for each latent variable from confirmatory factorial analysis, we consider a new approach of Condorcet cluster with number of groups fixed. In particular, we define new cost function to minimize, to find the optimal cluster assignment, fixing the number of groups. The proposed method is a valid competitor to the K-means method in terms of the optimal solution.

Key Words:K-means, Condorcet clustering, categorical data, social science

Universidad del Bío-Bío. Sistema de Bibliotecas - Chile

Universidad del Bío-Bío. Sistema de Bibliotecas - Chile

Índice general

1. Introducción **13**

2. Objetivos **17**
2.1. Objetivo General 17
2.2. Objetivos Específicos 17
2.3. Hipótesis 18

3. Metodología **19**
3.1. Método de Condorcet 19
3.1.1. Efecto Condorcet 20
3.1.2. Regla de agregación a la mayoría global 20
3.1.3. Ejemplo de aplicación del método 22
3.2. Métodos de agrupamiento 25
3.2.1. Algoritmos jerárquicos: 25
3.2.2. Agrupación particional: 26
3.3. K-means 27
3.3.1. Definición 28
3.3.2. Algoritmo 29
3.4. Análisis relacional 29
3.4.1. Metodología de agrupamiento de análisis relacional 30
3.4.2. La heurística del análisis relacional 33
3.4.3. Ejemplo de aplicación del método 35
3.4.4. Algoritmo de k-means basado en análisis relacional 37

4. Resultados **41**
4.1. Función de costo 41
4.2. Inercia intra-clases 43
4.3. Algoritmo propuesto 43
4.4. Comparación de métodos 46
4.5. Simulaciones 49

Universidad del Bío-Bío. Sistema de Bibliotecas - Chile

5. Discusión de resultados **57**

6. Conclusiones **59**

Referencias **64**

7. Anexo **65**
7.1. Código 65

Universidad del Bío-Bío. Sistema de Bibliotecas - Chile

Índice de figuras

4.1. Partición por Condorcet 47
4.2. Partición por K-means 47
4.3. Tasa de ajuste vs. número de particiones 48
4.4. Inercia intra-clase vs. número de particiones 49
4.5. Tasa de ajuste vs. diferentes grupos(α). Para distintas variables con n=15 51
4.6. Inercia vs. diferentes grupos(α). Para distintas variables con n=15 . 51
4.7. Tasa de ajuste vs. diferentes grupos(α). Para distintas variables con n=30 53
4.8. Inercia vs. diferentes grupos(α). Para distintas variables con n=30 . 53
4.9. Tasa de ajuste vs. diferentes grupos(α). Para distintas variables con n=50 55
4.10. Inercia vs. diferentes grupos(α). Para distintas variables con n=50 . 55

Universidad del Bío-Bío. Sistema de Bibliotecas - Chile

Universidad del Bío-Bío. Sistema de Bibliotecas - Chile

Índice de cuadros

3.1. Orden de preferencias 22
3.2. Datos del ejemplo 36
3.3. Matriz de Condorcet 36
3.4. Partición final 37

4.1. Tiempo de ejecución para diferentes variables, con n=15 50
4.2. Tiempo de ejecución para diferentes variables, con n=30 52
4.3. Tiempo de ejecución para diferentes variables, con n=50 54

Universidad del Bío-Bío. Sistema de Bibliotecas - Chile

Universidad del Bío-Bío. Sistema de Bibliotecas - Chile

Capítulo 1

Introducción

Una de las técnicas más utilizadas en el campo de la estadística, como también en las ciencias sociales y aprendizaje automático, son los métodos de agrupación; los cuales tienen por objetivo sintetizar y resumir grandes cantidades de datos. Sin embargo, el problema de la agrupación es la división de una población en grupos ("población": conjunto de elementos descritos por m atributos). El objetivo es construir una partición en la que los elementos de un clúster sean similares y los elementos de diferentes clústers sean distintos entre sí. No obstante, usualmente no es posible definir qué significa ser similar al igual que existe gran dificultad en la comparación de un resultado de agrupamiento con otro, produciendo un juicio generalmente subjetivo y dependiente de la aplicación. Existen diversas formas de definir una medida de adecuación para una partición determinada, de modo que la medida pueda servir al menos como función objetiva a optimizar en todas las particiones posibles (los algoritmos de agrupamiento se basan en hallar la partición que optimiza dicha función objetiva).

Marcotorchino y Pierre Michaud (1978,1985,1989) fueron los primeros en iniciar y desarrollar la teoría del análisis relacional, introduciendo el problema de

Universidad del Bío-Bío. Sistema de Bibliotecas - Chile

agregación de relaciones para resolver el problema de la votación (paradoja de Condorcet) y relacionarlo con el problema de similitud. Sus trabajos se basaron en investigaciones previas realizadas por Borda (1784) y Condorcet (1785).

Muchos autores consideraron imperfecto el trabajo de Condorcet, mientras que otros autores lo ignoraron. Pierre Michaud y Marcotorchino afirmaron que la solución propuesta por Condorcet fue malinterpretada. Sin embargo, confirmaron que la solución estaba incompleta. Estos autores, fueron los primeros en usar el álgebra pura en el sistema de votos (relación de orden y equivalencia), con el fin de resolver el problema de la paradoja o efecto de la votación, propusieron maximizar la función V(Y), correspondiente al número de votos que apoyan una opinión colectiva bajo restricciones.

Sea $N = \{v_1, ..., v_n\}$ el conjunto de n votantes y $X = \{x_1, ..., x_m\}$ el conjunto de m alternativas a elegir. Se define la matriz $C_{n\times n}$ como matriz de Condorcet, con elementos $c_{x_i x_j}$ correspondientes al número de votantes que prefiere x_i por sobre x_j o $x_i \succ x_j$. Se define también una matriz $Y_{n\times n}$ como la matriz de opiniones colectiva, que define las restricciones para la maximización del modelo. El modelo asociado a la regla de la mayoria global se define por:

$$MaxV(Y) = \sum_{x_i \neq x_j} (c_{x_i x_j} y_{x_i x_j} + \bar{c}_{x_i x_j} \bar{y}_{x_i x_j}),$$

$$y_{x_i x_j} \in \{0, 1\},$$

$$y_{x_i x_j} + y_{x_j x_i} = 1,$$

$$0 \leq y_{x_i x_j} + y_{x_j x_k} - y_{x_i x_k} \leq 1,$$

Universidad del Bío-Bío. Sistema de Bibliotecas - Chile

donde C es la matriz de Condorcet e Y es la matriz de opiniones colectiva. La segunda restricción representa la anti-simetría y la totalidad, mientras que la tercera restricción, presenta la transitividad.

En este trabajo estamos interesados en maximizar una nueva función de costo $V_\alpha(Y)$, fijando el número de grupos α. Luego dar su algoritmo heuristico. Finalmente, comparar los resultados de ambos métodos K-means y α-Condorcet sobre algunos valores fijos de α.

Universidad del Bío-Bío. Sistema de Bibliotecas - Chile

Universidad del Bío-Bío. Sistema de Bibliotecas - Chile

Capítulo 2

Objetivos

2.1. Objetivo General

Fijar el número de grupos en el método de Condorcet, con el fin de entregar su algoritmo y comparar los resultados de ambos métodos (K-means y α-Condorcet) para algunos valores fijos de α.

2.2. Objetivos Específicos

1. Extender el método de clasificación y agregación de datos Denominado Regla de Condorcet.

2. Definir una nueva función de costo fijando el número de grupos.

3. Comparar el método de K-means con el método α-Condorcet.

Universidad del Bío-Bío. Sistema de Bibliotecas - Chile

2.3. Hipótesis

1. Probar que la calidad de ajuste de α-Condorcet da mejores resultados que el método K-means.

Universidad del Bío-Bío. Sistema de Bibliotecas - Chile

Capítulo 3

Metodología

Según los objetivos de este trabajo, se propone estudiar el método de Condorcet como método de agrupamiento alternativo al de K-means.

3.1. Método de Condorcet

El *Método de Condorcet* es un sistema de votación (o preferencia) para elegir a una persona entre un grupo de candidatos. Este sistema de voto que siempre elige el ganador de Condorcet, es decir, el candidato al que los votantes prefieren ante el resto de ellos, cuando se les compara de uno en uno (comparaciones por pares). El mismo sistema satisface el criterio de Condorcet. Este criterio determina que el candidato comparado con el resto, es el preferido por el mayor número de votantes.

En la teoría de elección social, el método de Borda [4], fue el primer sistema de votación alternativo a la regla de la mayoría simple , que consiste en que cada votante le asigna un puntaje distinto a cada una de las alternativas, siendo el can-

Universidad del Bío-Bío. Sistema de Bibliotecas - Chile

didato ganador el que obtiene mayor cantidad de puntos.
Condorcet criticó la regla de Borda que dice que se puede elegir un candidato ganador sin que sea elegido por los votantes cómo una primera alternativa y propone a su vez, la regla de comparación por pares [5]. Más tarde, descubrió la llamada paradoja de la votación, denominada "efecto de Condorcet" [25], consiste en la aparición de ciclos en la estructura de preferencias, es decir, las preferencias sociales sobre un conjunto de alternativas pueden quedar no bien definidas.

3.1.1. Efecto Condorcet

Si la mayoría prefiere colectivamente a x antes que a y ($x \succ y$), y si la mayoría escoge a y antes de z ($y \succ z$); entonces la mayoría de los jueces prefieren colectivamente a z antes que x ($z \succ x$). Luego, se espera que la transitividad sea $x \succ z$. Sin embargo, estamos bajo el efecto Condorcet

$$x \succ y \succ z \succ x.$$

Para solucionar este problema, se propuso una regla que permite deducir una clasificación por preferencia con o sin la presencia de estos ciclos (efecto de Condorcet). P. Michaud la llamó regla de agregación a la mayoría global bajo las restricciones de Condorcet [22].

3.1.2. Regla de agregación a la mayoría global

Michaud y Marcotorchino [22],[20], fueron los primeros en usar el álgebra pura en el sistema de voto (relación de orden y equivalencia), con el fin de resolver el

Universidad del Bío-Bío. Sistema de Bibliotecas - Chile

problema de la paradoja o efecto de la votación. Proponen maximizar la función V(Y), correspondiente al número de votos que apoyan una opinión colectiva bajo restricciones. Sea $N = \{v_1, ..., v_n\}$ el conjunto de n votantes y $X = \{x_1, ..., x_m\}$ el conjunto de m alternativas a elegir. Se define la matriz $C_{n\times n}$ como matriz de Condorcet, con los elementos $c_{x_i x_j}$ correspondientes al número de votantes que prefiere x_i por sobre x_j o $x_i \succ x_j$. Se define también una matriz $Y_{n\times n}$ como la matriz de opiniones colectiva, se establece las restricciones para la maximización del modelo. El modelo asociado a la regla de la mayoria global se define por:

$$MaxV(Y) = \sum_{x_i \neq x_j} (c_{x_i x_j} y_{x_i x_j} + \bar{c}_{x_i x_j} \bar{y}_{x_i x_j}),$$

$$y_{x_i x_j} \in \{0, 1\},$$

$$y_{x_i x_j} + y_{x_j x_i} = 1,$$

$$0 \leq y_{x_i x_j} + y_{x_j x_k} - y_{x_i x_k} \leq 1,$$

donde C es la matriz de Condorcet e Y es la matriz de opiniones colectiva. La segunda restricción representa la anti-simetría y totalidad, mientras que la tercera restricción, presenta la transitividad.

En un voto, cuando no se presenta el problema de la simetria y no transitividad, la regla de Condorcet garantiza un solo ganador. Si, al menos uno de los dos problemas citados está presente, existe más de un ganador. Por ejemplo, si se presenta la simetría ($y_{x_i x_j} = y_{x_j x_i}$). Entonces, es posible obtener dos soluciones o más. Para dar una solución a este problema de empate, donde el número de votos apoyando una opinión (¿x_i es preferido sobre x_j?) sea igual al número de votos rechazando esta opinión, Michaud y Marcotorchino propusieron el siguiente

Universidad del Bío-Bío. Sistema de Bibliotecas - Chile

modelo:

$$MaxV(Y) = \sum_{x_i \neq x_j} (c_{x_i x_j} y_{x_i x_j} + \bar{c}_{x_i x_j} \bar{y}_{x_i x_j}),$$

$$y_{x_i x_j} \in \{0, 1\},$$

$$y_{x_i x_j} + y_{x_j x_i} \leq 1,$$

$$0 \leq y_{x_i x_j} + y_{x_j x_k} - y_{x_i x_k} \leq 1.$$

Sin embargo, el problema de la no existencia de una solución sigue presente bajo la no transitividad.

3.1.3. Ejemplo de aplicación del método

Sea m=4 candidatos (c_1, c_2, c_3, c_4) y n=5 votantes $(v_1, v_2, v_3, v_4, v_5)$. Cada votante opinó o realizó una clasificación de preferencia.

En forma matricial y según el orden de preferencias, se presenta la tabla que identifica la clasificación de los candidatos por los votantes.

Cuadro 3.1: Orden de preferencias

C	v_1	v_2	v_3	v_4	v_5
c_1	1	4	2	1	3
c_2	3	2	4	2	2
c_3	4	1	3	3	1
c_4	2	3	1	4	4

Orden de preferencia en forma matricial:

Universidad del Bío-Bío. Sistema de Bibliotecas - Chile

$$\begin{pmatrix} 1 & 4 & 2 & 1 & 3 \\ 3 & 2 & 4 & 2 & 2 \\ 4 & 1 & 3 & 3 & 1 \\ 2 & 3 & 1 & 4 & 4 \end{pmatrix}.$$

A continuación, la matriz de comparaciones por pares (Condorcet) está dada por:

$$C = \begin{pmatrix} 0 & 2 & 2 & 2 \\ 3 & 0 & 2 & 2 \\ 3 & 3 & 0 & 2 \\ 3 & 3 & 3 & 0 \end{pmatrix}.$$

Note que, por asunto de conveniencia, no se considera la diagonal. Ahora, se busca la matriz de las opiniones colectivas Y dado por:

Universidad del Bío-Bío. Sistema de Bibliotecas - Chile

$$y_{ij} = \begin{cases} 1 & , \text{ si } c_i \text{ gana a } c_j \\ 0 & , \text{ si } c_i \text{ pierde con } c_j. \end{cases}$$

Entonces la matriz Y es:

$$Y = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 \end{pmatrix}.$$

Una vez obtenida la matriz, se obtienen puntajes asociados a cada candidato según preferencias de votación por parte de ellos. Ésta se calcula sumando las filas de la matriz Y obteniendo:

$$Y = \begin{pmatrix} 0 \\ 1 \\ 2 \\ 3 \end{pmatrix}.$$

En conclusión, el ganador por la regla de Condorcet es el candidato (c_4).

Universidad del Bío-Bío. Sistema de Bibliotecas - Chile

3.2. Métodos de agrupamiento

Las técnicas de agrupación de clústers son un tipo de algoritmos de aprendizaje automático no supervisados. Estas técnicas no requieren entrenamiento de datos y forman grupos separados al reunir juntos en un grupo a observaciones similares. Para un conjunto de datos dado, el objetivo del análisis de conglomerados es definir un conjunto de conglomerados y asignar a cada uno de ellos las observaciones que, según alguna distancia o medida de similitud, están cerca entre sí, mientras que las observaciones entre los grupos están separados unos de otros. La bibliografía sobre análisis de conglomerados es muy extenso, algunos libros útiles son los siguientes: [1], [7], [10], [11], [13], [14] y [15]. Los racimos pueden considerarse nítidos o difusos. En el agrupamiento difuso, una observación puede pertenecer a más de un conglomerado con una probabilidad dada, mientras que al agrupar nítidamente, una observación pertenece a uno y solo a un grupo. Los algoritmos de agrupamiento pueden clasificarse en dos categorías: particionamiento y algoritmos jerárquicos.

3.2.1. Algoritmos jerárquicos:

Hay dos opciones de agrupamiento jerárquico, de arriba a abajo (divisivo) en el que comenzamos con todas las observaciones en un solo grupo. En el paso siguiente el clúster se divide en dos. Luego, los grupos se dividen continuamente hasta que hay tantos grupos como observaciones. Alternativamente, comenzamos con una observación por grupo y los grupos se fusionan hasta que todas las observaciones formen un solo grupo (agrupamiento aglomerativo). La agrupación jerárquica tiene la ventaja que, se puede aplicar el dendrograma, es fácil identificar el número

Universidad del Bío-Bío. Sistema de Bibliotecas - Chile

de clústers. Alternativamente, el criterio de varianza para seleccionar grupos [27]. sugirió este procedimiento, donde un par de clústers se fusionan en cada paso minimizando la varianza total dentro del conglomerado (son posibles otras funciones objetivo). Desafortunadamente, la agrupación jerárquica puede verse muy afectada por valores atípicos y también puede llevar mucho tiempo para grandes conjuntos de datos.

3.2.2. Agrupación particional:

En este método de agrupamiento, las observaciones se dividen en k grupos, donde el número de clústers, k, lo define el usuario.

K-means

k-Means [9],[19], es una de las técnicas de agrupación más populares, es rápida y fácil de implementar. Una vez definido el número de conglomerados k, este método elige al azar k observaciones del conjunto de datos y los utiliza como media inicial. El algoritmo no garantiza convergencia al óptimo global y como suele ser un algoritmo rápido, es común ejecutarlo varias veces con diferentes condiciones de inicio. Este método puede verse muy afectado por valores atípicos. Se logra un método más robusto reemplazando las medias con medianas.

Spectral clustering

Esta técnica de agrupamiento se basa en la teoría de grafos [8], este enfoque se utiliza para identificar comunidades de nodos en un gráfico basado en los bordes que la conectan. Como método de aprendizaje automático, la agrupación espectral

Universidad del Bío-Bío. Sistema de Bibliotecas - Chile

se popularizó en [26],[24]. La agrupación espectral se basa en los valores propios y vectores propios de la matriz laplaciana derivada de una matriz de similitud utilizada para realizar una reducción de dimensión. Entonces, un algoritmo de agrupamiento clásico, como k-means, se aplica en menos dimensiones.

DBSCAN

Este algoritmo de agrupación fue propuesto en [6]. Es un algoritmo basado en la densidad: las observaciones se agrupan según lo cerca que están el uno de otros. Es decir, un punto pertenecerá a un clúster si está cerca de muchos puntos de ese grupo.

Este enfoque de agrupación requiere la especificación de dos parámetros: determinar el radio dentro del cual los puntos deben estar en un grupo; el segundo especifica el número mínimo de puntos necesarios para formar un racimo. DBSCAN, es uno de los algoritmos de agrupación en clúster más utilizados. Es resistente a la presencia de valores atípicos (por lo tanto, se puede utilizar para la detección de valores atípicos). También tiene la ventaja que no requiere especificar el número de clústers. de antemano y funciona bien con formas arbitrarias de grupos. Por otra parte, es sensible a las especificaciones de los parámetros.

3.3. K-means

El algoritmo k-means es uno de los algoritmos de agrupamiento más utilizados diseñado para agrupar datos numéricos, donde cada agrupamiento contiene un centro llamado centroide. El objetivo de este algoritmo es encontrar K centros de clústers cuya suma de los cuadrados de las distancias de los puntos al centro más

Universidad del Bío-Bío. Sistema de Bibliotecas - Chile

cercano sea mínima[9], [19].

El término "k-means" fue utilizado por primera vez por James MacQueen [19], aunque la idea se remonta a Hugo Steinhaus en 1957. El algoritmo estándar fue propuesto por primera vez por Stuart Lloyd en 1957 como una técnica para modulación por impulsos codificados, aunque no se publicó fuera de los laboratorios Bell [18] .En [12] proporcionan una historia detallada de K-means junto con descripciones de varias variaciones.

3.3.1. Definición

Sea O un conjunto de n objetos, $D = \{x_1, x_2, ..., x_n\}$ el conjunto de vectores en $\mathcal{R}^m$ de características de los objetos. K-means construye una partición de las observaciones en k conjuntos ($k \leq n$) a fin de minimizar la suma de los cuadrados dentro de cada grupo (inercia intra-clases). Si $S_1, S_2, ..., S_k$ son k grupos disjuntos de O definimos el centroide del grupo i como:

$$\mu(S_i) = \frac{1}{|S_i|} \sum_{j \in S_i} x_j. \tag{3.1}$$

Lo que se quiere es minimizar la suma de los cuadrados dentro de cada grupo (inercia intra-clases). Es decir,

$$min_S = \sum_{i=1}^{k} \sum_{x_j \in S_i} ||x_j - \mu_i||^2, \tag{3.2}$$

donde μ_i es el centro de gravedad de la nube del clúster S_i.

Universidad del Bío-Bío. Sistema de Bibliotecas - Chile

3.3.2. Algoritmo

El procedimiento estándar de k-means se resume a continuación:

1. Inicializar la matriz de centroides. $M = [\mu(S_1), \mu(S_2), ..., \mu(S_k)]$.

2. Asignar a cada objeto de O el grupo más cercano S_i con respecto a una métrica d de $\mathcal{R}^m$, es decir para $j = 1, ..., n$.

$$o_j \in S_l, \text{ si } d(x_j, \mu(S_l)) \leq d(x_j, \mu(S_i))$$

$$\forall i = 1, ..., k.$$

3. Calcular la nueva matriz de centroides aplicando (2) a la nueva partición.

4. Repetir los pasos 2 y 3 hasta que no haya ningún cambio en cada grupo.

Algunas propiedades y variantes del algoritmo k-means se discuten en [28], [10].

3.4. Análisis relacional

La teoría de análisis relacional fue propuesta y desarrollada por Marcotorchino y Pierre Michaud. introdujeron la agregación de relaciones para resolver el problema de la votación (la paradoja de Condorcet) y lo relacionaron con el problema de la similitud. Sus investigaciones se basaron en trabajos anteriores de Borda [4] y Condorcet[5].

El análisis relacional se define como un problema de optimización bajo restricciones lineales del criterio de Condorcet. La representación matemática es la siguente [2], [17].

Universidad del Bío-Bío. Sistema de Bibliotecas - Chile

3.4.1. Metodología de agrupamiento de análisis relacional

Para agrupar un conjunto de datos $\mathcal{P}$ compuesto por n observaciones $(O_1, O_2, ..., O_n)$ descrito por m variables $(V^1, V^2, ..., V^m)$, primero comenzamos transformando cada columna V^k en una matriz relacional C^k con término general $c^k_{ii'}$ definida por:

$$c^k_{ii'} = \begin{cases} 1 & \text{si } O_i \text{ y } O_{i'} \text{ tienen la misma modalidad de la variable } V^k, \\ 0 & \text{En otro caso.} \end{cases}$$

Este término representa la similitud entre las observaciones O_i y $O_{i'}$ con respecto a la variable V^k. Una vez construidas todas las m matrices C^k, construimos una matriz de relacional global C llamada "matriz de Condorcet" de término general $c_{ii'}$ que representa la similitud global de O_i y $O_{i'}$, con respecto al conjunto completo de las m variables: $c_{ii'} = \sum_{k=1}^{m} C^k_{ii'}$. Esta similitud global tiene la llamada "propiedad de similitud máxima" definida por: $c_{ii'} \leq \mathcal{M}_{ii'}\ \forall O_i, O_{i'}$,donde $\mathcal{M}_{ii'} = Min(c_{ii}, c_{i'i'})$ es la máxima similitud posible entre las dos observaciones O_i y $O_{i'}$.

Usando la similitud global $c_{ii'}$ y la máxima similitud posible $\mathcal{M}_{ii'}$ entre O_i y $O_{i'}$, los autores, Marcotorchino y Pierre Michaud [23] definen su disimilitud como:

$$\bar{c}_{ii'} = \mathcal{M}_{ii'} - c_{ii'}. \tag{3.3}$$

Dos observaciones estarán, a priori, en el mismo grupo de la partición final esperada si su similitud es mayor que su disimilitud, es decir, $c_{ii'} \geq \bar{c}_{ii'}$. La partición final requerida estará representada por una matriz X cuadrada y binaria de N x

Universidad del Bío-Bío. Sistema de Bibliotecas - Chile

N con término general $x_{ii'}$ definido como:

$$x_{ii'} = \begin{cases} 1 & \text{si } O_i \text{ y } O_{i'} \text{ están en el mismo grupo de la partición final,} \\ 0 & \text{En otro caso.} \end{cases}$$

Esta partición se obtendrá maximizando el criterio $\mathcal{C}(X)$ de Condorcet definido a continuación:

$$\mathcal{C}(X) = \sum_{i=1}^{n}\sum_{i'=1}^{n}(c_{ii'}x_{ii'} + \bar{c}_{ii'}\bar{x}_{ii'}), \tag{3.4}$$

donde,

$$\bar{x}_{ii'} = 1 - x_{ii'}. \tag{3.5}$$

Usando las expresiones (3.3) y (3.5), el criterio $\mathcal{C}(X)$ se puede reescribir como:

$$\mathcal{C}(X) = \sum_{i=1}^{n}\sum_{i'=1}^{n}(2c_{ii'} - M_{ii'})x_{ii'} + \sum_{i=1}^{n}\sum_{i'=1}^{n}\bar{c}_{ii'}. \tag{3.6}$$

Como el segundo miembro de la suma de la expresión (3.6) es una constante, deducimos que maximizar el criterio de Condorcet es equivalente a maximizar la siguiente función:

$$\mathcal{C}'(X) = \sum_{i=1}^{n}\sum_{i'=1}^{n}(c_{ii'} - \frac{\mathcal{M}_{ii'}}{2})x_{ii'}. \tag{3.7}$$

La función de costo del criterio $\mathcal{C}'(X)$ será positiva cuando la similitud $c_{ii'}$ entre

Universidad del Bío-Bío. Sistema de Bibliotecas - Chile

dos observaciones O_i y $O_{i'}$ es mayor o igual a la mitad de su similitud máxima posible. Esta condición es a veces muy difícil de alcanzar, especialmente cuando el número de variables (o descriptores) es muy alto en comparación con el número de observaciones, es decir, m>n.

Como el objetivo de la tarea de agrupamiento es resumir la cantidad de datos en estructuras más simples, para evitar este problema, una solución consiste en relajar la función de costo relacionada con el criterio de agrupamiento. Para alcanzar ese objetivo es suficiente reemplazar el coeficiente $\frac{1}{2}$ de $\mathcal{M}_{ii'}$ por un parámetro α tal que $0 < \alpha < \frac{1}{2}$. La nueva formulación del criterio $\mathcal{C}'(X)$ será entonces:

$$\mathcal{C}'(X) = \sum_{i=1}^{n}\sum_{i'=1}^{n}(c_{ii'} - \alpha\mathcal{M}_{ii'})x_{ii'}. \tag{3.8}$$

Por tanto, la formulación matemática del problema de agrupamiento del análisis relacional es:

$$\max_{X} \mathcal{C}'(X),$$

bajo las restricciones:

$$\begin{cases} x_{ii'} \in \{0,1\} & \forall(\mathrm{O_i}, \mathrm{O_{i'}}) \in \mathcal{P}^2, \quad \text{(binaria)} \\ x_{ii} = 1 & \forall \mathrm{O_i} \in \mathcal{P}, \quad \text{(reflexividad)} \\ x_{ii'} - x_{i'i} = 0 & \forall(\mathrm{O_i}, \mathrm{O_{i'}}) \in \mathcal{P}^2, \quad \text{(simetria)} \\ x_{ii'} - x_{i'i''} - x_{ii''} \leqslant 1 & \forall(\mathrm{O_i}, \mathrm{O_{i'}}, \mathrm{O_{i''}}) \in \mathcal{P}^3, \quad \text{(transitividad)} \end{cases}$$

Universidad del Bío-Bío. Sistema de Bibliotecas - Chile

3.4.2. La heurística del análisis relacional

La solución exacta del problema anterior se puede obtener mediante técnicas de programación lineal. Cuando la población estudiada es relativamente pequeña. Pero, en la práctica, el conjunto de datos puede exceder cientos de miles o millones de observaciones. Esta situación lleva a utilizar una heurística, para obtener la mejor solución más cercana a la solución exacta, en un tiempo razonable de proceso. A continuación, damos la descripción de la heurística que fue utilizada por el análisis relacional [3], [16], [17].

Fase 1.

Este paso consiste en inicializar el proceso de agrupación construyendo una primera partición. Para construir esta primera partición, construimos progresivamente sus clústers según las operaciones descritas:

1. Inicialización: tomamos aleatoriamente una primera observación que constituye el primer grupo de la partición desconocida.

2. Tomamos una observación O_i perteneciente $\mathcal{P}$ y calcular su enlace $\mathcal{L}_{iv}$, con todas las agrupaciones existentes V.

$$\mathcal{L}_{iv} = \sum_{i' \in V} \mathcal{L}_{ii'},$$

donde el enlace $\mathcal{L}_{ii'}$ entre O_i y $O_{i'}$:

$$\mathcal{L}_{ii'} = C_{ii'} - \alpha \mathcal{M}_{ii'}.$$

Universidad del Bío-Bío. Sistema de Bibliotecas - Chile

Esta observación se asigna al grupo que tiene el mayor vínculo estrictamente positivo. Si todos los enlaces son negativos, creamos un nuevo clúster para incluir esta nueva observación.

3. Repita este proceso hasta que todas las observaciones de la población $\mathcal{P}$ hayan sido asignadas a un grupo.

Fase 2.

Al final del primer paso, obtenemos una partición con varios clústers.

1. Fusionando dos clústers: Tomamos, ahora, los grupos uno tras otro y calculamos el enlace $\mathcal{L}_{\mathcal{V}\mathcal{V}'}$ de cada clúster $\mathcal{V}$ con todos los demás $\mathcal{V}'$.

$$\mathcal{L}_{\mathcal{V}\mathcal{V}'} = \mathcal{A}_{\mathcal{V}\mathcal{V}'} - \alpha\mathcal{M}_{\mathcal{V}\mathcal{V}'},$$

donde los acuerdos $\mathcal{A}_{\mathcal{V}\mathcal{V}'}$ entre dos clúster es:

$$\mathcal{A}_{\mathcal{V}\mathcal{V}'} = \sum_{i\in\mathcal{V}}\sum_{i'\in\mathcal{V}'} c_{ii'},$$

y los desacuerdos $\bar{\mathcal{A}}_{\mathcal{V}\mathcal{V}'}$ entre dos clúster es:

$$\bar{\mathcal{A}}_{\mathcal{V}\mathcal{V}'} = \sum_{i\in\mathcal{V}}\sum_{i'\in\mathcal{V}'} \bar{c}_{ii'},$$

y la máxima similitud posible $\mathcal{M}_{\mathcal{V}\mathcal{V}'}$ entre dos clúster es:

$$\mathcal{M}_{\mathcal{V}\mathcal{V}'} = \sum_{i\in\mathcal{V}}\sum_{i'\in\mathcal{V}'} \mathcal{M}_{ii'}.$$

Universidad del Bío-Bío. Sistema de Bibliotecas - Chile

Luego, fusionaremos los clúster que tengan el mejor vínculo (valor positivo estricto más alto). Esto debe llevarse a cabo siempre que exista la posibilidad de mejorar el criterio $\mathcal{C}'(X)$.

2. Transferir una observación de un clúster a otro. Ahora, tomamos las observaciones de cada grupo y calculamos el enlace $\mathcal{L}_{iv}$ (Fase 1) de cada observación O_i con los otros grupos $\mathcal{V}$. Si una observación tiene un mejor vínculo con otro grupo que el suyo, entonces esta observación se transfiere a este nuevo grupo. Esto se llevará a cabo siempre que se mejore el criterio $\mathcal{C}'(X)$.

Cuando no es posible la transferencia de una observación, volvemos al paso de fusión para ver si es posible mejorar el criterio de Condorcet fusionando otros clústers. Estos cuatro pasos se aplicarán, hasta que no se produzcan más mejoras del criterio.

3.4.3. Ejemplo de aplicación del método

Ejemplo de [17]. Supongamos que la población estudiada $\mathcal{P}$ está compuesta por siete observaciones $(O_1, O_2, ..., O_7)$ que tienen tres variables cualitativas V_1, V_2, V_3. El conjunto de datos se presenta en la tabla 3.2.

Después de la transformación de las tres variables cualitativas en su matriz relacional que representa la matriz de Condorcet C. Representado en la tabla 3.3.

Universidad del Bío-Bío. Sistema de Bibliotecas - Chile

Cuadro 3.2: Datos del ejemplo

	V_1	V_2	V_3
O_1	1	1	1
O_2	1	1	1
O_3	1	2	2
O_4	2	2	2
O_5	2	2	2
O_6	3	2	3
O_7	3	3	3

Cuadro 3.3: Matriz de Condorcet

	O_1	O_2	O_3	O_4	O_5	O_6	O_7
O_1	3	3	1	0	0	0	0
O_2	3	3	1	0	0	0	0
O_3	1	1	3	2	2	1	0
O_4	0	0	2	3	3	1	0
O_5	0	0	2	3	3	1	0
O_6	0	0	1	1	1	3	2
O_7	0	0	0	0	0	2	3

Como el número de variables medidas en esta población es igual a tres, también se representa la "máxima similitud posible" que puede ocurrir entre dos observaciones O_i y $O_{i'}$. Entonces podemos deducir que la diferencia global entre esas observaciones es $\bar{c}_{ii'} = 3 - c_{ii'}$. La matriz cuadrada binaria X, que representa la partición final obtenida de la población $\mathcal{P}$ tiene el siguiente término general:

$$x_{ii'} = \begin{cases} 1 & \text{si } c_{ii'} \geq \bar{c}_{ii'}, \\ 0 & \text{En otro caso.} \end{cases}$$

Finalmente, se alcanza el óptimo con la siguente partición X.

Universidad del Bío-Bío. Sistema de Bibliotecas - Chile

Cuadro 3.4: Partición final

	O_1	O_2	O_3	O_4	O_5	O_6	O_7
O_1	1	1	0	0	0	0	0
O_2	1	1	0	0	0	0	0
O_3	0	0	1	1	1	0	0
O_4	0	0	1	1	1	0	0
O_5	0	0	1	1	1	0	0
O_6	0	0	0	0	0	1	1
O_7	0	0	0	0	0	1	1

El valor de criterio de Condorcet correspondiente es: $\mathcal{C} = 131$, donde los grupos son:

- Grupo 1 : O_1, O_2
- Grupo 2 : O_3, O_4, O_5
- Grupo 3 : O_6, O_7.

3.4.4. Algoritmo de k-means basado en análisis relacional

Lazhar Labiod [16] propuso un enfoque de análisis relacional para el algoritmo k-means, presentando la siguiente inercia entre clases:

$$I_Y = \frac{1}{2}\sum_{i=1}^{N}\sum_{i'=1}||x_i - x_{i'}||^2\frac{y_{ii'}}{y_i}. \tag{3.9}$$

Las características relacionales son $y_{ii'}^k = \sum_k \chi_{ik}\chi_{i'k}$, donde χ_{ik} y $\chi_{i'k}$ son variables dummy de la matriz particionada. El término x_i representa el i-ésimo vector del conjunto de datos X, y k es una característica en particular.

Universidad del Bío-Bío. Sistema de Bibliotecas - Chile

En lugar de la distancia euclidiana utilizada en el algoritmo clásico de k-means, esta formulación permite usar otra distancia o similitud / disimilitud o cualquier otra función que se observe $D_{ii'}$. Entonces, reemplazando la distancia euclidiana al cuadrado entre dos observaciones (i, i') por $D_{ii'}$. El problema de agrupamiento tiene la siguiente forma relacional:

$$\min_{y} I_w(y), \tag{3.10}$$

donde

$$I_w(Y) = \frac{1}{2} \sum_{i=1}^{N} \sum_{i'=1}^{N} D_{ii'} \frac{y_{ii'}}{y_{i.}}, \tag{3.11}$$

sujeto a:

$$\begin{cases} Y & \text{Matriz de relación de equilavencia de IxI} \\ \\ \sum_{i=1}^{N} \sum_{i=1}^{N} \frac{y_{ii'}}{y_{i.} y_{.i'}} = K. \end{cases} \tag{3.12}$$

Para gestionar diferentes tipos de datos, existe una formulación relacional general del algoritmo k-means propuesto por los mismos autores utilizando diferentes criterios para cada tipo de datos. El enfoque de agrupamiento unificado se ilustra a continuación:

$$\min_{Y} \frac{1}{2} \sum_{i=1}^{N} \sum_{i'=1}^{N} K_{ii'} \frac{y_{ii'}}{y_{io}},$$

bajo las restricciones anteriores(3.12),

Universidad del Bío-Bío. Sistema de Bibliotecas - Chile

donde K es la matriz de afinidad, que podría ser una matriz de distancia, de similitud o de diferencias. Esta matriz se define de la siguiente manera:

$$K_{ii'} = \begin{cases} D_{ii'} \in \mathcal{R}^+ & \text{(Distancia)} \rightarrow \text{tipo 1;} \\ S_{ii'} \in \mathcal{R}^+ & \text{Similitud} \rightarrow \text{tipo 1;} \\ D_{ii'} - S_{ii'} \in \mathcal{R} & \text{(Distancia-Similitud)} \rightarrow \text{tipo 2;} \\ S_{ii'} - D_{ii'} \in \mathcal{R} & \text{(Similitud-Distancia)} \rightarrow \text{tipo 2.} \end{cases}$$

Entonces, hay dos tipos de criterios:

1. Tipo 1 (criterio con coste positivo o cero): Este criterio se basa en la distancia o similitud. El costo del criterio es siempre positivo o cero.

2. Tipo 2 (criterio con costo real): El criterio del costo real se refiere a los enfoques que utilizan simultáneamente una distancia y una medida de similitud. Según el perfil de cada muestra, el coste de este criterio puede ser positivo e incluso negativo. Estos criterios tienen el costo de Condorcet.

Universidad del Bío-Bío. Sistema de Bibliotecas - Chile

Universidad del Bío-Bío. Sistema de Bibliotecas - Chile

Capítulo 4

Resultados

En el capítulo se propone una versión mejorada del método de Condorcet. Este nuevo método propone introducir un parámetro α que permita controlar el número de grupos. Luego, se plantea un algoritmo heurístico, para obtener la mejor solución más cercana a la solución exacta, en un tiempo razonable de proceso.

Se ha realizado un estudio de desempeño integral para evaluar nuestro método. En esta sección, describimos los experimentos y sus resultados. Ejecutamos nuestro algoritmo en el conjunto de datos Felinos obtenidos en [21] y datos simulados en R-project para probar el rendimiento de nuestro algoritmo frente a K-means.

A continuación, presentamos el primer resultado importante.

4.1. Función de costo

Se define una nueva función de costo, agregando un nuevo parámetro α que controla el número de clústers.

Universidad del Bío-Bío. Sistema de Bibliotecas - Chile

Definición 1.

Sean S_k, $k = 1, \cdots, \alpha$, α agrupaciones. Entonces, se define una nueva función de costo por,

$$g(S;\alpha) = \sum_{k=1}^{\alpha} \sum_{i,j} \left(c_{ij} S_{ik} S_{jk} + \overline{c}_{ij} (\overline{S}_{ik} S_{jk} + S_{ik} \overline{S}_{jk}) \right),$$

donde $S = (S_{ik})_{m \times \alpha}$ con $S_{ik} = 1_{\{x_i \in S_k\}}$.

El modelo que se presenta a continuación permite agrupar los objetos por semejanza, fijando a priori el número exacto de los grupos. Entonces, se obtiene la partición requerida resolviendo el sistema (4.1).

$$\mathcal{P}' \quad \begin{cases} \max g(S_1, \cdots, S_\alpha; \alpha) \\ S_{ik} \in \{0,1\}, \\ \sum_{i=1}^{m} S_{ik} \geq 1, \qquad k = 1, \cdots, \alpha \\ \sum_{k=1}^{\alpha} S_{ik} = 1. \end{cases} \tag{4.1}$$

Usando los resultados de Michaud y Marcotorchino [22],[20], tenemos los siguientes resultados:

Teorema 1. Sea $P = \{x_1, \cdots, x_m\}$ el conjunto las observaciones. Entonces, existe al menos una partición de P ($P = \cup_{k=1}^{\alpha} S_k$ con $S_i \cap S_j = \varnothing$, $\forall \alpha \leq m$), que maximize $\mathcal{P}'$.

Teorema 2. Se supone que el problema en estudio no presenta el efecto de Condorcet. Entonces, existe una única partición de P que maximize $\mathcal{P}'$.

Universidad del Bío-Bío. Sistema de Bibliotecas - Chile

4.2. Inercia intra-clases

Ahora, nos enfocamos en la inercia intra-clases o suma de cuadrados dentro de los grupos. Sea $(S_k)_{k=\{1,\cdots,\alpha\}}$ una partición del conjunto de las observaciones S. Luego, constuimos una nube de puntos, denotado $N(S)$, tal que $N(S) \in \mathbb{R}^n$ donde cada dimensión correponde a una categoría de la variable v_j, $j = 1, \cdots, n$. Se define $N(S) = \{(A_i, \mu_i) : i \in S\}$ como una nube de puntos de masa, donde A_i es la coordenada de la observación x_i y μ_i su masa correspondiente. En general, la suma de los cuadrados dentro de los grupos está dada por:

$$I_w = \sum_{k}^{\alpha} \sum_{i \in S_k} \mu_i ||A_i - G_k||^2, \tag{4.2}$$

donde G_k es el centro de gravedad de la nube del clúster S_k.

La nueva expresión relacional de la inercia intra-clases, con número de clústers α fijo, es dada por:

$$I_w = \frac{1}{M} \left(p - \sum_{i=1}^{m} \sum_{j=1}^{m} \hat{c}_{ij} \sum_{k=1}^{\alpha} \frac{S_{ik} S_{jk}}{|S_k|} \right), \tag{4.3}$$

donde $|S_k|$ es la cardinalidad de S_k.

4.3. Algoritmo propuesto

El algoritmo propuesto similar al dado por [17]. Se encontraron algunas desventajas al tomar un elemento al azar, este puede quedar mal clasificado, por ejemplo. Además, computacionalmente es muy costoso al ser un algoritmo heurístico.

Universidad del Bío-Bío. Sistema de Bibliotecas - Chile

El algoritmo de agrupación de α-Condorcet se ilustra mediante algunos pasos. Dada una base de datos D de n puntos en $\mathbb{R}^n$ y la partición $S_1, \cdots, S_n$ tal que $S_k = \{x_k\}$, $k = 1, \cdots, n$.

De manera similar al algoritmo dado por [17], calculamos el siguiente paso.

1. Al principio, se considera que cada observación forma un nuevo grupo. Entonces decimos que la colección de todas las observaciones forman una partición de S, y escribimos $\cup_{i=1}^{m} = S$.

2. Seleccionamos la observación con mayor vínculo con otros clúster , teniendo en cuenta que el enlace de cualquier grupo se define como:

$$L_{S_k S_{k'}} = \sum_{j \in S_{k'}} \sum_{i \in S_k} (2C_{ij} - M).$$

Los clústers se fusionarán cuando el valor de su enlace sea el más alto.

En este paso se modifica el algoritmo y see define el vínculo entre los grupos como:

$$L_{S_k S_{k'}} = \sum_{j \in S_{k'}} \sum_{i \in S_k} C_{ij},$$

donde C_{ij}, es la matriz de Condorcet.

Universidad del Bío-Bío. Sistema de Bibliotecas - Chile

Al definir el enlace de esta forma, tiene la misma estructura que el enlace definido por [17]. Debido a que M es constante, se define el enlace de esta manera para trabajar con enlace entre grupos positivos. Por la estructura del algoritmo, pero se puede trabajar con el enlace definido por [17].

a) Buscamos la posición del mayor enlace entre todos los grupos (puede ser por fila o columna). En este caso, se toma la columna. Lo denotamos como S_k.

b) Se define $a = L_{S_k S_{k'}}$ como el mayor enlace de la fila en la posición S_k. Luego, se elimina el valor de a y se define $b = L_{S_{k''} S_{k'}}$ con el mayor enlace de la columna en la posición $S_{k'}$. Por otro lado, se define otro elemento "di" como la frecuencia del elemento a.

c) Agregamos unas condiciones:

1) **Si a>b :** Como "a" alcanza el máximo enlace entre los objetos S_k y $S_{k'}$. Luego, se elimina el valor del enlace en la posición $(S_k, S_{k'})$. Ahora se observan los elementos que se encuentran en la posición $S_{k'}$. Se sabe que el objeto con el máximo enlace se encuentra en la posición $S_{k''}$ cuyo valor es "b". Como a>b, entonces el elemento en la posición $(S_k, S_{k'})$ alcanza el valor máximo del enlace. En consecuencia se agrupan.

2) **Si a=b :** Se buscar la posición donde "di" sea la mas frecuente. Luego, se aplica el criterio de Condorcet. Agrupamos aquellos elementos que maximicen el criterio de Condorcet.

3) **Si a<b :** Se aplica el mismo criterio de a>b.

Universidad del Bío-Bío. Sistema de Bibliotecas - Chile

3. Repetimos el proceso.

4. Este proceso se detiene cuando se identifican los α grupos.

4.4. Comparación de métodos

En esta sección, describimos los experimentos y sus resultados. Ejecutamos nuestro algoritmo en conjuntos de datos felinos obtenidos en [21]. Para probar su rendimiento de agrupación de α-Condorcet frente al algoritmo de K-means.

Nuestros algoritmos se implementaron en R- project.

En esta sección, comparamos α-Condorcet con el algoritmo de K-means. El algoritmo α-Condorcet se implementa de acuerdo con la descripción en este proyecto, y el algoritmo de k-means se realiza estrictamente de acuerdo con el algoritmo implementado en R- project. Se compara la calidad de la partición mediante la tasa de ajuste (Michaud,1989). Que está dada por:

$$Ta = 2\,\text{máx}\, g(S;\alpha)/(m^2 n).$$

En estudios previos [21], [22]. El método de agregación de similitud (Condorcet) nos brinda una solución óptima de 4 grupos. Esta solución está más cerca a la clasificación reconocida por los zoólogos por especies y por género(Figura 4.1).

Por otra parte, en la partición en 4 grupos aplicando el algoritmo de K-means, se observa que ciertas especies pertenecen a más de un grupo y que no concuerda con la clasificación reconocida por los zoólogos. (Figura 4.2)

Universidad del Bío-Bío. Sistema de Bibliotecas - Chile

Figura 4.1: Partición por Condorcet

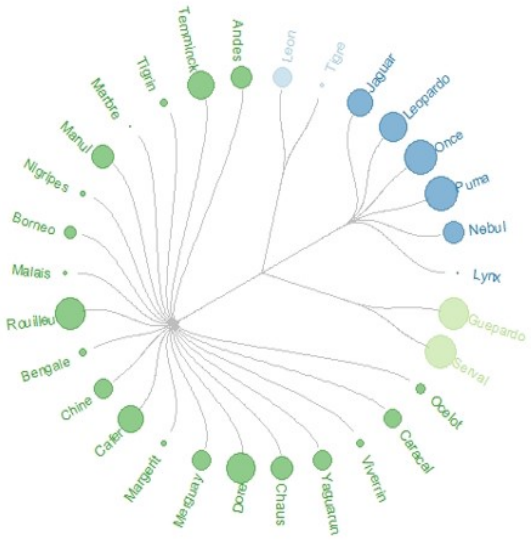
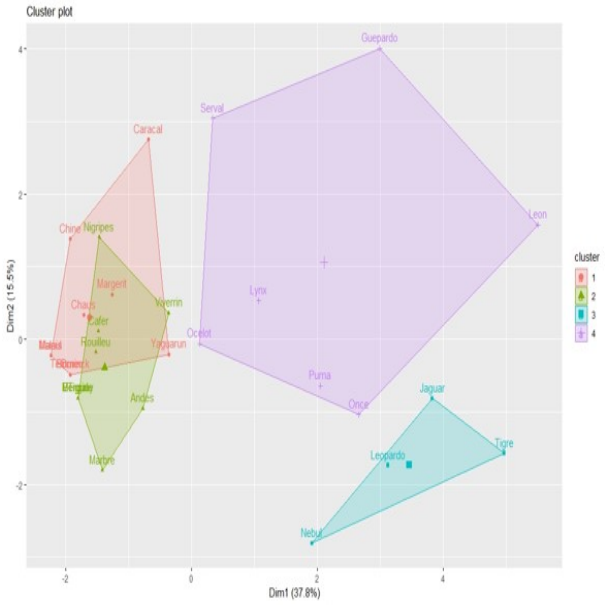


Figura 4.2: Partición por K-means



A continuación, usamos los α-Condorcet con el algoritmo de K-means para agrupar el conjunto de datos de felinos en diferentes números de grupos. Para cada número fijo de agrupaciones, se compararon la calidad de la partición mediante la

Universidad del Bío-Bío. Sistema de Bibliotecas - Chile

tasa de ajuste e inercia.

La figura 4.3, contrasta la calidad de ajuste de agrupación mediante la tasa de ajuste, en el conjunto de datos Felinos. Podemos concluir que el rendimiento de nuestro algoritmo es superior al rendimiento de K-means para un grupo fijo α.

La figura 4.4, contrasta la calidad de ajuste de agrupación mediante la inercia, en el conjunto de datos Felinos. No podemos concluir que el rendimiento de nuestro algoritmo es superior al rendimiento de K-means para un grupo fijo α.

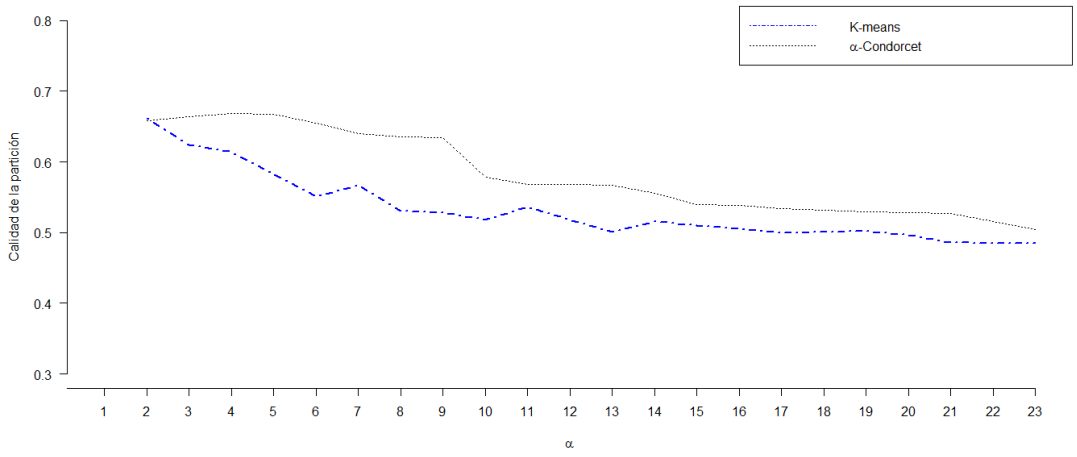


Figura 4.3: Tasa de ajuste vs. número de particiones

Universidad del Bío-Bío. Sistema de Bibliotecas - Chile

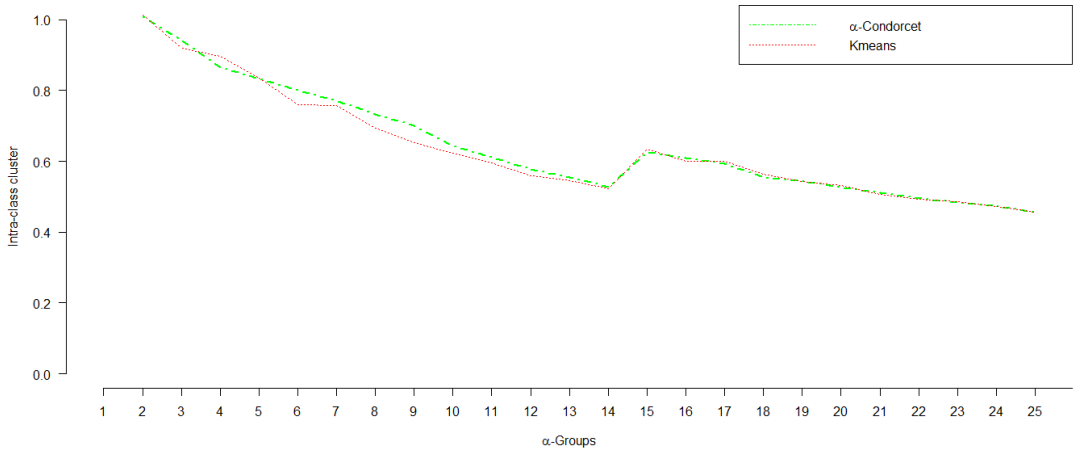


Figura 4.4: Inercia intra-clase vs. número de particiones

4.5. Simulaciones

En esta sección, describimos los experimentos y sus resultados. Ejecutamos nuestro algoritmo en conjuntos de datos simulados en R-project. Para probar su rendimiento de agrupación de α-Condorcet frente al algoritmo de K-means.

Se compara la calidad de la partición, se considera la tasa de ajuste y la nueva expresión relacional de la inercia intra-clases, con número de clústers α fijo(α-Condorcet).

A continuación, en el cuadro 4.1, se compara α-Condorcet con el algoritmo de K-means para un conjunto de datos simulado con n=15 en diferentes números de clústers. Se observan diferentes variables con n=15 fijo. Luego, se comparan el tiempo de ejecución para la tasa de ajuste e inercia. El tiempo de ejecución es considerado desde 2-Cordorcet hasta 13-Corcorcet.

Universidad del Bío-Bío. Sistema de Bibliotecas - Chile

Cuadro 4.1: Tiempo de ejecución para diferentes variables, con n=15

	Condorcet	**K-means**
n	15	15
variables	tiempo de ejecución	tiempo de ejecución
2	tasa de ajuste=0.017m, inercia=0.025m	tasa de ajuste=0.01m, inercia=0.016m
3	tasa de ajuste=0.016m, inercia=0.021m	tasa de ajuste=0.012m, inercia=0.015m
4	tasa de ajuste=0.1m, inercia=0.1m	tasa de ajuste=0.01m, inercia=0.015m
5	tasa de ajuste=0.1m, inercia=0.1m	tasa de ajuste=0.01m, inercia=0.015m

posteriormente, se compara α-Condorcet con el algoritmo de K-means para agrupar el conjunto de datos simulado con n=15, para distintas variables, en diferentes número de grupos. Para cada número fijo de agrupaciones, se compararon la calidad de partición mediante la tasa de ajuste (Figura 4.4) e inercia (Figura 4.5).

Al observar las figuras 4.4 y 4.5, se contrasta la calidad de ajuste de agrupación en el conjunto de datos. Podemos concluir que el rendimiento de nuestro algoritmo es superior al rendimiento de K-means para un grupo fijo α. Tanto para la tasa de ajuste e inercia.

Universidad del Bío-Bío. Sistema de Bibliotecas - Chile

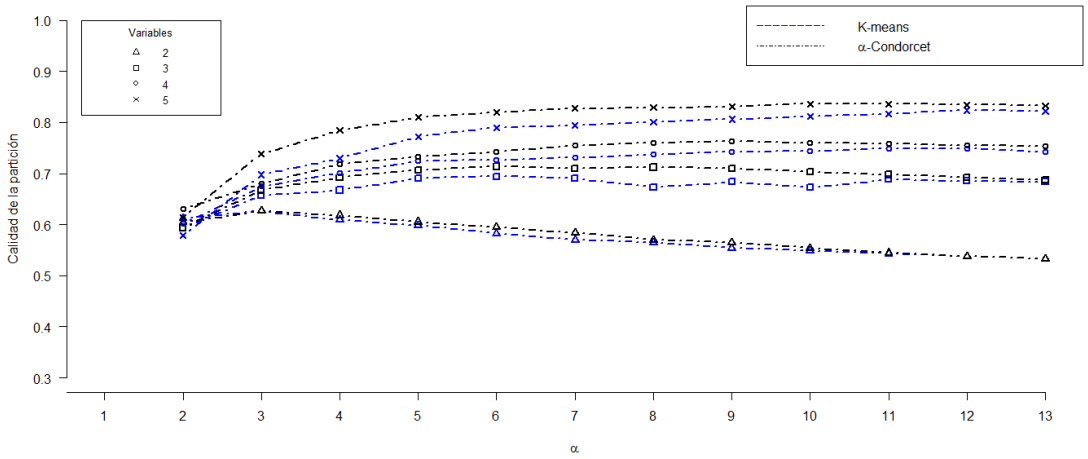


Figura 4.5: Tasa de ajuste vs. diferentes grupos(α). Para distintas variables con n=15

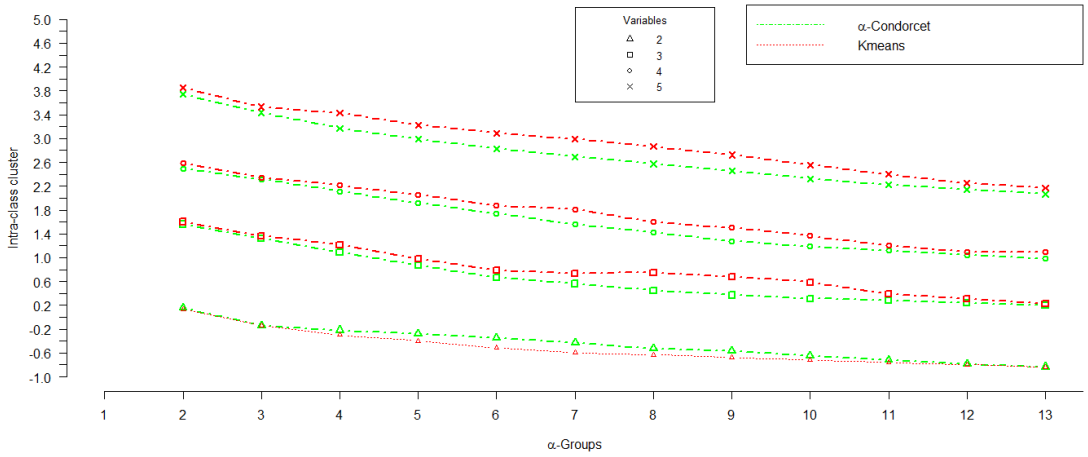


Figura 4.6: Inercia vs. diferentes grupos(α). Para distintas variables con n=15

A continuación, en el cuadro 4.2, se compara α-Condorcet con el algoritmo de K-means para un conjunto de datos simulado con n=30 en diferentes números de clústers. Se observan diferentes variables para n=30 fijo. Luego, se comparan el

Universidad del Bío-Bío. Sistema de Bibliotecas - Chile

tiempo de ejecución para la tasa de ajuste e inercia. El tiempo de ejecución es considerado desde 2-Cordorcet hasta 28-Corcorcet.

Cuadro 4.2: Tiempo de ejecución para diferentes variables, con n=30

	Condorcet	**K-means**
n	30	30
Variables	tiempo de computo	tiempo de computo
2	tasa de ajuste=0.24m, inercia=0.25m	tasa de ajuste=0.012m, inercia=0.02m
3	tasa de ajuste=0.4m, inercia=0.41m	tasa de ajuste=0.012m, inercia=0.02m
4	tasa de ajuste=1.42m, inercia=1.43m	tasa de ajuste=0.011m, inercia=0.02m
5	tasa de ajuste=4.1m, inercia=4.13m	tasa de ajuste=0.012m, inercia=0.02m

posteriormente, se compara α-Condorcet con el algoritmo de K-means para agrupar el conjunto de datos simulado con n=30, para distintas variables, en diferentes números de grupos. Para cada número fijo de agrupaciones, se compararon la calidad de partición mediante la tasa de ajuste (Figura 4.7) e inercia (Figura 4.8).

Al observar las figuras 4.7, se contrasta la calidad de ajuste de agrupación mediante la tasa de ajuste en el conjunto de datos. Podemos concluir que el rendimiento de nuestro algoritmo es superior al rendimiento de K-means para un grupo fijo α.

La figura 4.8, contrasta la calidad de ajuste de agrupación mediante la inercia en el conjunto de datos . No podemos concluir que el rendimiento de nuestro algoritmo es superior al rendimiento de K-means para un grupo fijo α. Para valores positivos la inercia es mejor para Condorcet en comparación con K-means.

Universidad del Bío-Bío. Sistema de Bibliotecas - Chile

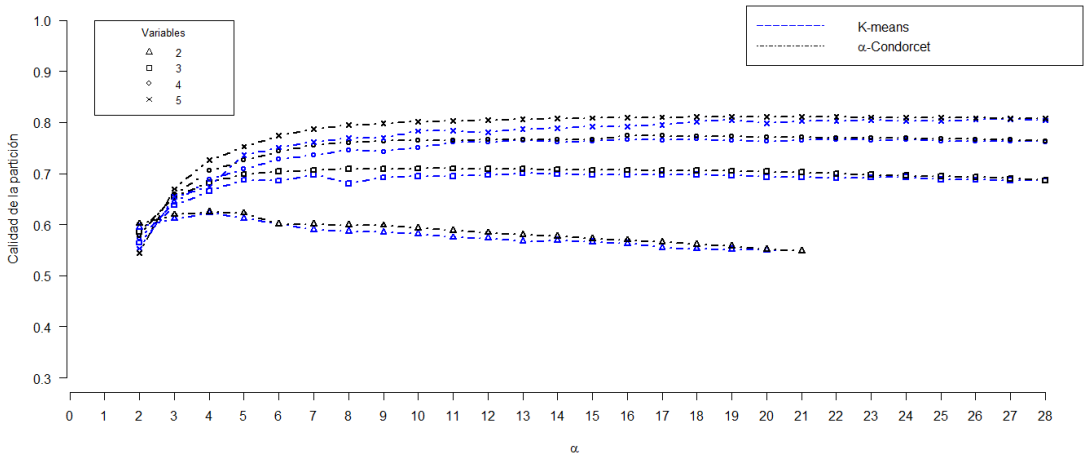


Figura 4.7: Tasa de ajuste vs. diferentes grupos(α). Para distintas variables con n=30

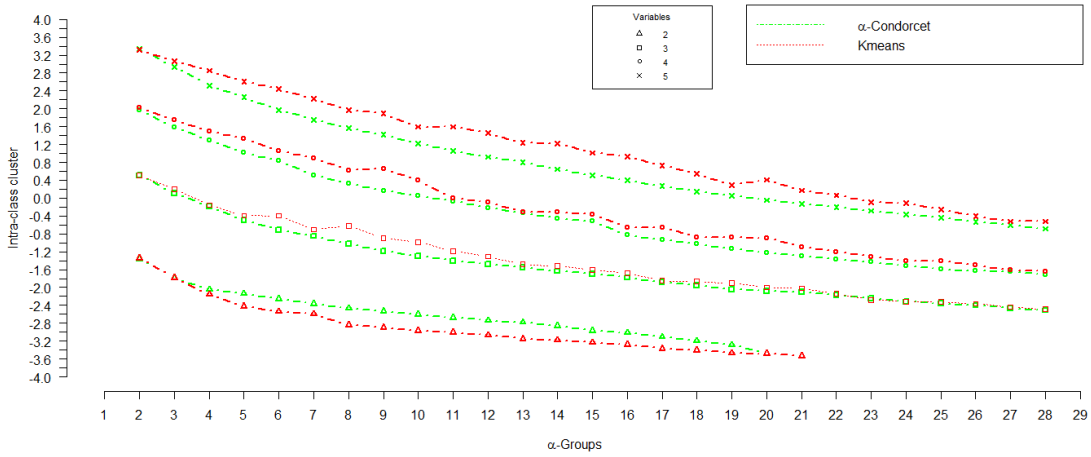


Figura 4.8: Inercia vs. diferentes grupos(α). Para distintas variables con n=30

A continuación, en el cuadro 4.3, se compara α-Condorcet con el algoritmo de K-means para un conjunto de datos simulado para n=50 en diferentes números de clústers. Se observan diferentes variables con n=50 fijo. Luego, se comparan

Universidad del Bío-Bío. Sistema de Bibliotecas - Chile

el tiempo de ejecución para la tasa de ajuste e inercia. El tiempo de ejecución es considerado desde 2-Cordorcet hasta 47-Corcorcet.

Cuadro 4.3: Tiempo de ejecución para diferentes variables, con n=50

	Condorcet	**K-means**
n	50	50
Variables	tiempo de computo	tiempo de computo
2	tasa de ajuste=3.8m, inercia=3.9m	tasa de ajuste=0.02m, inercia=0.03m
3	tasa de ajuste=5.8m, inercia=5.9m	tasa de ajuste=0.016m, inercia=0.02m
4	tasa de ajuste=10.5m, inercia=10.6m	tasa de ajuste=0.016m, inercia=0.03m
5	tasa de ajuste=16.1m, inercia=16.2m	tasa de ajuste=0.016m, inercia=0.028m

posteriormente, se compara α-Condorcet con el algoritmo de K-means para agrupar el conjunto de datos simulado con n=50, para distintas variables, en diferentes números de grupos. Para cada número fijo de agrupaciones, se compararon la calidad de partición mediante la tasa de ajuste (Figura 4.9) e inercia (Figura 4.10).

Al observar las figuras 4.9, se contrasta la calidad de ajuste de agrupación mediante la tasa de ajuste en el conjunto de datos. Podemos concluir que el rendimiento de nuestro algoritmo es superior al rendimiento de K-means para un grupo fijo α.

La figura 4.10, contrasta la calidad de ajuste de agrupación mediante la inercia en el conjunto de datos. No podemos concluir que el rendimiento de nuestro algoritmo es superior al rendimiento de K-means para un grupo fijo α.

Universidad del Bío-Bío. Sistema de Bibliotecas - Chile

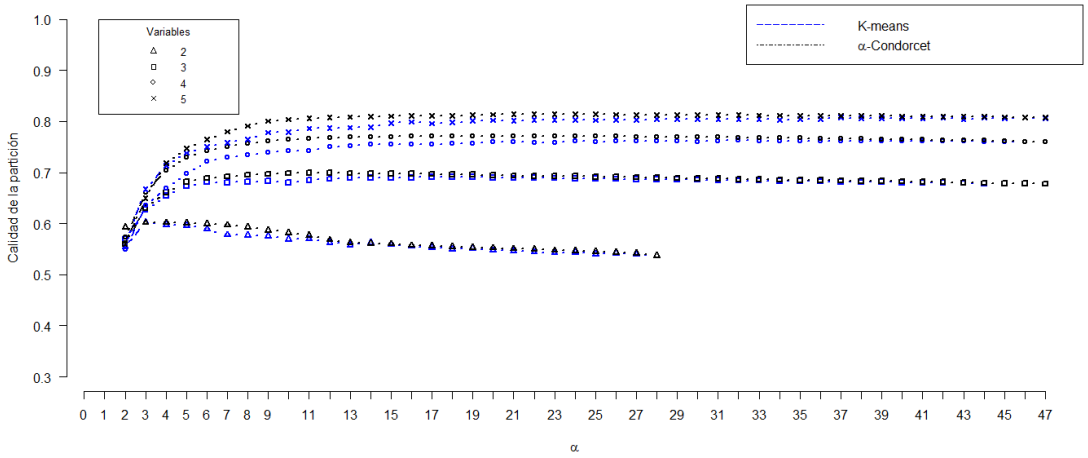


Figura 4.9: Tasa de ajuste vs. diferentes grupos(α). Para distintas variables con n=50

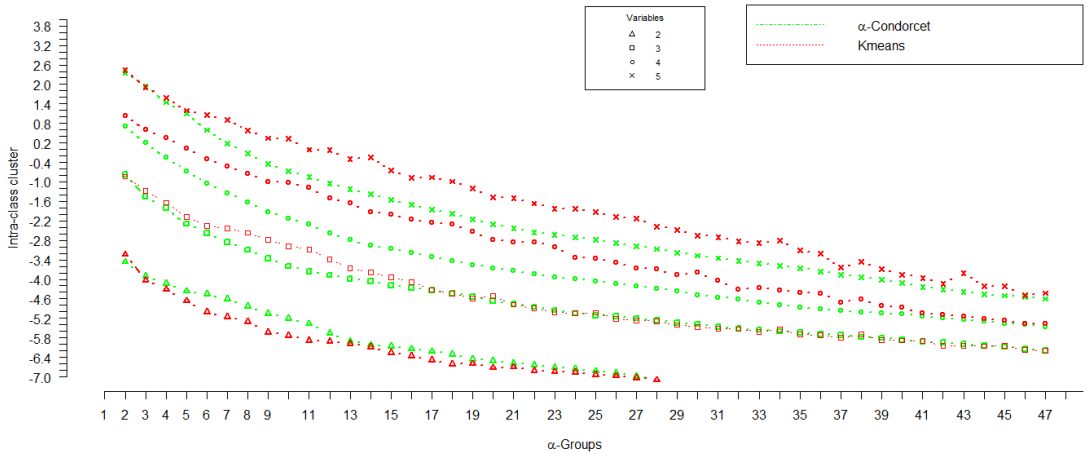


Figura 4.10: Inercia vs. diferentes grupos(α). Para distintas variables con n=50

Universidad del Bío-Bío. Sistema de Bibliotecas - Chile

Universidad del Bío-Bío. Sistema de Bibliotecas - Chile

Capítulo 5

Discusión de resultados

Tras comparar los resultados de α-Condertet con K-means en la base de datos felinos, se cumple la hipótesis sobre α-Condertet es superior para un α dado, para la tasa de ajuste. Sin embargo, cuando se trata de la inercia, esta no resulta ser superior.

Al aplicar α-Condertet, a datos simulados nos da lo mismo resultados que la base de datos Felinos. Con $n \geq 30$, la inercia empieza a dar peores resultados.

La gran limitación de este estudio es que no es generalizables a una población mayor. Por limitación computacional. α-Condertet, se puede comparar con el algoritmo de K-means, con un pequeño conjunto de datos entregando la solución óptima. Dado que el algoritmo de α-Condertet es un algoritmo heurístico, no se puede garantizar que para un conjunto de datos mayor, alcance la solución óptima.

Universidad del Bío-Bío. Sistema de Bibliotecas - Chile

Universidad del Bío-Bío. Sistema de Bibliotecas - Chile

Capítulo 6

Conclusiones

El método de clasificación Condorcet es muy poco conocido, y a diferencia de sus pares entrega una solución óptima de clasificación. Su proceso se basa en datos que se encuentran ya relacionados entre sí, obteniendo una matriz de comparación de a pares, para luego maximizar el modelo mediante procesos de iteración y programación lineal cumpliendo restricciones de simetría y transitividad. Una gran ventaja de este método es la ya mencionada, de entregar una solución óptima. Es por esta razón que se utilizará con el fin de lograr una clasificación óptima.

Los métodos de agrupación son fundamentales para muchas ramas de la estadística, así como para las ciencias sociales y el aprendizaje automático, entre otras. En las últimas décadas, ha habido un creciente debate sobre el mejor método de agrupación. Para muchos fenómenos se conoce la clasificación por expertos en algunas área de interés. Lo resultados expuesto, en este proyecto extienden el método de Condorcet a un número dado de grupos.

El modelo que se presentó permite agrupar los objetos por semejanza, fijando a

Universidad del Bío-Bío. Sistema de Bibliotecas - Chile

priori el número exacto de los grupos (α-Condorcet). Dado su complejidad computacional, se aplicó un algoritmo heurístico, dando buenos resultados al comparlo con el algoritmo de K-means.

Universidad del Bío-Bío. Sistema de Bibliotecas - Chile

Bibliografía

[1] C. C. Aggarwal and C. K. Reddy. Data clustering. *Algorithms and applications. Chapman&Hall/CRC Data mining and Knowledge Discovery series, Londra*, 2014.

[2] J. Ah-Pine and J.-F. Marcotorchino. Overview of the relational analysis approach in data-mining and multi-criteria decision making. *Web intelligence and intelligent agents*, pages 325–346, 2010.

[3] H. Benhadda and F. Marcotorchino. L'analyse relationnelle pour la fouille de grandes bases de données. *Revue des Nouvelles Technologies de l'Information*, pages 149–167, 2007.

[4] J. d. Borda. Mémoire sur les élections au scrutin. *Histoire de l'Academie Royale des Sciences pour 1781 (Paris, 1784)*, 1784.

[5] M. d. Condorcet. Essay on the application of analysis to the probability of majority decisions. *Paris: Imprimerie Royale*, 1785.

[6] M. Ester, H.-P. Kriegel, J. Sander, X. Xu, et al. A density-based algorithm for discovering clusters in large spatial databases with noise. In *Kdd*, volume 96, pages 226–231, 1996.

Universidad del Bío-Bío. Sistema de Bibliotecas - Chile

[7] B. S. Everitt, S. Landau, M. Leese, and D. Stahl. Cluster analysis 5th ed, 2011.

[8] M. Fiedler. Algebraic connectivity of graphs. *Czechoslovak mathematical journal*, 23(2):298–305, 1973.

[9] E. W. Forgy. Cluster analysis of multivariate data: efficiency versus interpretability of classifications. *biometrics*, 21:768–769, 1965.

[10] G. Gan, C. Ma, and J. Wu. *Data clustering: theory, algorithms, and applications*. SIAM, 2020.

[11] C. Hennig, M. Meila, F. Murtagh, and R. Rocci. *Handbook of cluster analysis*. CRC Press, 2015.

[12] A. K. Jain and R. C. Dubes. *Algorithms for clustering data*. Prentice-Hall, Inc., 1988.

[13] A. Kassambara. *Practical guide to cluster analysis in R: Unsupervised machine learning*, volume 1. Sthda, 2017.

[14] L. Kaufman and P. J. Rousseeuw. *Finding groups in data: an introduction to cluster analysis*, volume 344. John Wiley & Sons, 2009.

[15] R. S. King. *Cluster analysis and data mining: An introduction*. Stylus Publishing, LLC, 2015.

[16] L. Labiod. *Contribution au formalisme relationnel des classifications simultanées de deux ensembles*. PhD thesis, Paris 6, 2008.

Universidad del Bío-Bío. Sistema de Bibliotecas - Chile

[17] M. Lebbah, Y. Bennani, N. Grozavu, and H. Benhadda. Relational analysis for clustering consensus. *Machine Learning*, pages 46–60, 2010.

[18] S. Lloyd. Least square quantization in pcm. bell telephone laboratories paper. published in journal much later: Lloyd, sp: Least squares quantization in pcm. *IEEE Trans. Inform. Theor.(1957/1982)*, 18, 1957.

[19] J. MacQueen et al. Some methods for classification and analysis of multivariate observations. In *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, volume 1, pages 281–297. Oakland, CA, USA, 1967.

[20] F. Marcotorchino. Liaison analyse factorielle-analyse relationnelle, 1989.

[21] F. Marcotorchino and P. Michaud. Agregation de similarites en classification automatique. *Revue de statistique appliquée*, 30(2):21–44, 1982.

[22] P. Michaud. Agrégation à la majorité ii: analyse du résultat d'un vote. 1985.

[23] P. Michaud and F. Marcotorchino. Modèles d'optimisation en analyse des données relationnelles. *Mathématiques et Sciences humaines*, 67:7–38, 1979.

[24] A. Y. Ng, M. I. Jordan, Y. Weiss, et al. On spectral clustering: Analysis and an algorithm. *Advances in neural information processing systems*, 2:849–856, 2002.

[25] A. O'Connor, F. Arago, et al. *Œuvres de Condorcet: Éloges, t. I-II*, volume 2. Firmin Didot frères, 1847.

[26] J. Shi and J. Malik. Normalized cuts and image segmentation. *IEEE Transactions on pattern analysis and machine intelligence*, 22(8):888–905, 2000.

Universidad del Bío-Bío. Sistema de Bibliotecas - Chile

[27] J. H. Ward Jr. Hierarchical grouping to optimize an objective function. *Journal of the American statistical association*, 58(301):236–244, 1963.

[28] R. Xu and D. Wunsch. *Clustering*, volume 10. John Wiley & Sons, 2008.

Universidad del Bío-Bío. Sistema de Bibliotecas - Chile

Capítulo 7

Anexo

7.1. Código

A continuación presentamos el código correspondiente a nuestro trabajo. La programación está en R-project:

```
library(gtools)
alfa<-function(data,alpha){
C<-matrix(0,ncol=dim(data)[1],nrow=dim(data)[1])
for(i in 1:dim(data)[1]){
for(j in 1:dim(data)[1]){
if(i<j){
C[i,j]=sum(as.numeric(data[i,]==data[j,]))
}else
C[i,j]=sum(as.numeric(data[i,]==data[j,]))
}
C[i,i]=dim(data)[2]
}
C_prima<-dim(data)[2]-C
M<-dim(data)[2]
enlace<-C
enlace<-enlace-diag(diag(enlace))
enlace2<-enlace
```

Universidad del Bío-Bío. Sistema de Bibliotecas - Chile


```
grupos<-list()
A<-dim(data)[1]
B<-1:dim(data)[1]
suma<-1

while(A>alpha && suma!=0){
k<-numeric(dim(data)[1])
for(j in 1:dim(data)[1]){
k[j]<-max(enlace[,j])
}
ab<-unlist(grupos)
if(A==dim(data)[1]){
ini<-which.max(k)
}else{
k[ab]=0
ini<-which.max(k)
}

posicion<-which.max(enlace[ini,])
a<-enlace[ini,posicion]
di<-which(enlace[ini,]==a)
enlace[ini,posicion]=0
posicion_columna<-which.max(enlace[,posicion])
b<-enlace[posicion_columna,posicion]
enlace[ini,posicion]<-a

if(a<b){
k<-numeric(dim(data)[1])
for(j in 1:dim(data)[1]){
k[j]<-max(enlace[,j])
}
ini<-which.max(k)
posicion<-which.max(enlace[ini,])
a<-enlace[ini,posicion]
di<-which(enlace[ini,]==a)
enlace[ini,posicion]=0
posicion_columna<-which.max(enlace[,posicion])
b<-enlace[posicion_columna,posicion]
enlace[ini,posicion]<-a
}
```

Universidad del Bío-Bío. Sistema de Bibliotecas - Chile


```
if(a>b){
if(length(di)==1){
if(dim(data)[1]==A){
grupos[[1]]<-c(ini,posicion)
ab<-unlist(grupos)
largo<-length(B[-ab])
A<-length(grupos)+largo
enlace[,posicion]=0
enlace[posicion,]=0
}else{
grupos[[length(grupos)+1]]<-c(ini,posicion)
enlace[,posicion]=0
enlace[posicion,]=0
ab<-unlist(grupos)
dup<-ab[!(!duplicated(ab) & rev(!duplicated(rev(ab))))]
if(length(dup)>=2){
if(length(dup)==4){
ak<-which(sapply(grupos, function(e) is.element(dup[1], e)))
grupos[[ak[1]]]<-unique(unlist(grupos[ak]))
grupos[[ak[2]]]<-NULL
largo<-length(B[-ab])
A<-length(grupos)+largo
ab<-unlist(grupos)
dup<-ab[!(!duplicated(ab) & rev(!duplicated(rev(ab))))]
ak<-which(sapply(grupos, function(e) is.element(dup[1], e)))
grupos[[ak[1]]]<-unique(unlist(grupos[ak]))
grupos[[ak[2]]]<-NULL
largo<-length(B[-ab])
A<-length(grupos)+largo
}else{
az<-which(sapply(grupos, function(e) is.element(dup[1], e)))
grupos[[az[1]]]<-unique(unlist(grupos[az]))
grupos[[az[length(az)]]]<-NULL
largo<-length(B[-ab])
A<-length(grupos)+largo
}
} #termina if(lengt(dup)>=2)
largo<-length(B[-ab])
```

Universidad del Bío-Bío. Sistema de Bibliotecas - Chile


```
A<-length(grupos)+largo
} #termina el else de  if(dim(data)[1]==A)
#terminar el if(
}else{
af2<-c(ini,di)
combinaciones<-list()
for(ag in 2:length(af2)){
combi <- combinations(length(af2), ag, af2)
for(an in 1:dim(combi)[1]){
combinaciones[[length(combinaciones)+1]]<-combi[an,]
}
}
Calidad_c<-rep(0,length(combinaciones))
for(kp in 1:length(combinaciones)){
grupos2<-grupos
grupos2[[length(grupos2)+1]]<-combinaciones[[kp]]
ab<-unlist(grupos2)
dup<-ab[!(!duplicated(ab) & rev(!duplicated(rev(ab))))]
if(length(dup)>=2){
if(length(dup)==4){
ak<-which(sapply(grupos2, function(e) is.element(dup[1], e)))
grupos2[[ak[1]]]<-unique(unlist(grupos2[ak]))
grupos2[[ak[2]]]<-NULL
largo<-length(B[-ab])
A<-length(grupos2)+largo
ab<-unlist(grupos2)
dup<-ab[!(!duplicated(ab) & rev(!duplicated(rev(ab))))]
ak<-which(sapply(grupos2, function(e) is.element(dup[1], e)))
grupos2[[ak[1]]]<-unique(unlist(grupos2[ak]))
grupos2[[ak[2]]]<-NULL
largo<-length(B[-ab])
A<-length(grupos2)+largo
}else{
az<-which(sapply(grupos2, function(e) is.element(dup[1], e)))
grupos2[[az[1]]]<-unique(unlist(grupos2[az]))
grupos2[[az[length(az)]]]<-NULL
largo<-length(B[-ab])
A<-length(grupos2)+largo
}
} #termina length(dup)>=2
```

Universidad del Bío-Bío. Sistema de Bibliotecas - Chile


```
ab2<-unlist(grupos2)
ac2<-B[-ab2]
z2<-list()
z2<-grupos2
for(ip in 1:length(ac2)){
z2[[length(grupos2)+ip]]<-ac2[ip]
}
class_C<-numeric()
for(n in 1:length(z2)){
class_C[z2[[n]]]<-n
}
Y_c<-matrix(0,ncol=dim(data)[1],nrow=dim(data)[1])
for(i in 1:dim(data)[1]){
for(j in 1:dim(data)[1]){
Y_c[i,j]=ifelse(class_C[i]==class_C[j],1,0)

}}
Y_c1<-(1-Y_c)
C1<-(dim(data)[2]-C)
Calidad_c[kp]<-sum(C*Y_c+C1*Y_c1)
} # termina el for kp
max_calidad<-which.max(Calidad_c)
posi_calidad<-unlist(combinaciones[max_calidad])
if(length(which(posi_calidad==ini))==0){
ini<-posi_calidad[1]
di<-posi_calidad[-1]
grupos[[length(grupos)+1]]<-c(ini,di[1])
ab<-unlist(grupos)
dup<-ab[!(!duplicated(ab) & rev(!duplicated(rev(ab))))]
if(length(dup)>=2){
if(length(dup)==4){
ak<-which(sapply(grupos, function(e) is.element(dup[1], e)))
grupos[[ak[1]]]<-unique(unlist(grupos[ak]))
grupos[[ak[2]]]<-NULL
largo<-length(B[-ab])
A<-length(grupos)+largo
ab<-unlist(grupos)
dup<-ab[!(!duplicated(ab) & rev(!duplicated(rev(ab))))]
ak<-which(sapply(grupos, function(e) is.element(dup[1], e)))
grupos[[ak[1]]]<-unique(unlist(grupos[ak]))
```

Universidad del Bío-Bío. Sistema de Bibliotecas - Chile


```
grupos[[ak[2]]]<-NULL
largo<-length(B[-ab])
A<-length(grupos)+largo
}else{
az<-which(sapply(grupos, function(e) is.element(dup[1], e)))
grupos[[az[1]]]<-unique(unlist(grupos[az]))
grupos[[az[length(az)]]]<-NULL
largo<-length(B[-ab])
A<-length(grupos)+largo}
}else{
largo<-length(B[-ab])
A<-length(grupos)+largo
}
enlace[,di[1]]=0
enlace[di[1],]=0
di<-di[-1]
if(length(di)!=0){
for(j in 1:length(di)){
if(A>alpha){
enlace[,di[j]]=0
enlace[di[j],]=0
grupos[[length(grupos)+1]]<-c(ini,di[j])
ab<-unlist(grupos)
dup<-ab[!(!duplicated(ab) & rev(!duplicated(rev(ab))))]
if(length(dup)>=2){
if(length(dup)==4){
ak<-which(sapply(grupos, function(e) is.element(dup[1], e)))
grupos[[ak[1]]]<-unique(unlist(grupos[ak]))
grupos[[ak[2]]]<-NULL
largo<-length(B[-ab])
A<-length(grupos)+largo
ab<-unlist(grupos)
dup<-ab[!(!duplicated(ab) & rev(!duplicated(rev(ab))))]
ak<-which(sapply(grupos, function(e) is.element(dup[1], e)))
grupos[[ak[1]]]<-unique(unlist(grupos[ak]))
grupos[[ak[2]]]<-NULL
largo<-length(B[-ab])
A<-length(grupos)+largo
}else{
az<-which(sapply(grupos, function(e) is.element(dup[1], e)))
```

Universidad del Bío-Bío. Sistema de Bibliotecas - Chile


```
grupos[[az[1]]]<-unique(unlist(grupos[az]))
grupos[[az[length(az)]]]<-NULL
largo<-length(B[-ab])
A<-length(grupos)+largo
}
} ####termina el if(length(dup)>=2)
} #termina if(A>alpha)
} #termina el for
} ##termina el length(di)!=0
posi_calidad<-numeric
}else{ ### el else de if(length(which(posi_calidad==ini))==0)
if(length(posi_calidad)==length(c(ini,di))){
if(length(di)>1){
if(dim(data)[1]==A){
grupos[[1]]<-c(ini,posicion)
ab<-unlist(grupos)
largo<-length(B[-ab])
A<-length(grupos)+largo
enlace[,posicion]=0
enlace[posicion,]=0
}else{
grupos[[length(grupos)+1]]<-c(ini,posicion)
ab<-unlist(grupos)
dup<-ab[!(!duplicated(ab) & rev(!duplicated(rev(ab))))]
if(length(dup)>=2){
if(length(dup)==4){
ak<-which(sapply(grupos, function(e) is.element(dup[1], e)))
grupos[[ak[1]]]<-unique(unlist(grupos[ak]))
grupos[[ak[2]]]<-NULL
largo<-length(B[-ab])
A<-length(grupos)+largo
ab<-unlist(grupos)
dup<-ab[!(!duplicated(ab) & rev(!duplicated(rev(ab))))]
ak<-which(sapply(grupos, function(e) is.element(dup[1], e)))
grupos[[ak[1]]]<-unique(unlist(grupos[ak]))
grupos[[ak[2]]]<-NULL
largo<-length(B[-ab])
A<-length(grupos)+largo
}else{
az<-which(sapply(grupos, function(e) is.element(dup[1], e)))
```

Universidad del Bío-Bío. Sistema de Bibliotecas - Chile


```
grupos[[az[1]]]<-unique(unlist(grupos[az]))
grupos[[az[length(az)]]]<-NULL
largo<-length(B[-ab])
A<-length(grupos)+largo}
}else{
largo<-length(B[-ab])
A<-length(grupos)+largo
}
enlace[,posicion]=0
enlace[posicion,]=0
}
di2<-which(enlace[ini,]==a)
for(j in 1:length(di2)){
if(A>alpha){
posicion<-which.max(enlace[ini,])
a2<-enlace[ini,posicion]
enlace[ini,posicion]=0
posicion_columna<-which.max(enlace[,posicion])
b2<-enlace[posicion_columna,posicion]
enlace[ini,posicion]<-a2
if(a2>=b2){
enlace[,posicion]=0
enlace[posicion,]=0
grupos[[length(grupos)+1]]<-c(ini,posicion)
ab<-unlist(grupos)
dup<-ab[!(!duplicated(ab) & rev(!duplicated(rev(ab))))]
if(length(dup)>=2){
if(length(dup)==4){
ak<-which(sapply(grupos, function(e) is.element(dup[1], e)))
grupos[[ak[1]]]<-unique(unlist(grupos[ak]))
grupos[[ak[2]]]<-NULL
largo<-length(B[-ab])
A<-length(grupos)+largo
ab<-unlist(grupos)
dup<-ab[!(!duplicated(ab) & rev(!duplicated(rev(ab))))]
ak<-which(sapply(grupos, function(e) is.element(dup[1], e)))
grupos[[ak[1]]]<-unique(unlist(grupos[ak]))
grupos[[ak[2]]]<-NULL
largo<-length(B[-ab])
A<-length(grupos)+largo
```

Universidad del Bío-Bío. Sistema de Bibliotecas - Chile


```
}else{
az<-which(sapply(grupos, function(e) is.element(dup[1], e)))
grupos[[az[1]]]<-unique(unlist(grupos[az]))
grupos[[az[length(az)]]]<-NULL
largo<-length(B[-ab])
A<-length(grupos)+largo
}
} #termina el if(a>=b)
}
} #termina if(A>alpha)
} #termina el for
#aca terminar el if(length)>=2
}else{
if(dim(data)[1]==A){
grupos[[1]]<-c(ini,posicion)
ab<-unlist(grupos)
largo<-length(B[-ab])
A<-length(grupos)+largo
enlace[,posicion]=0
enlace[posicion,]=0
}else{
grupos[[length(grupos)+1]]<-c(ini,posicion)
enlace[,posicion]=0
enlace[posicion,]=0
ab<-unlist(grupos)
dup<-ab[!(!duplicated(ab) & rev(!duplicated(rev(ab))))]
if(length(dup)>=2){
if(length(dup)==4){
ak<-which(sapply(grupos, function(e) is.element(dup[1], e)))
grupos[[ak[1]]]<-unique(unlist(grupos[ak]))
grupos[[ak[2]]]<-NULL
largo<-length(B[-ab])
A<-length(grupos)+largo
ab<-unlist(grupos)
dup<-ab[!(!duplicated(ab) & rev(!duplicated(rev(ab))))]
ak<-which(sapply(grupos, function(e) is.element(dup[1], e)))
grupos[[ak[1]]]<-unique(unlist(grupos[ak]))
grupos[[ak[2]]]<-NULL
largo<-length(B[-ab])
A<-length(grupos)+largo
```

Universidad del Bío-Bío. Sistema de Bibliotecas - Chile


```
}else{
az<-which(sapply(grupos, function(e) is.element(dup[1], e)))
grupos[[az[1]]]<-unique(unlist(grupos[az]))
grupos[[az[length(az)]]]<-NULL
largo<-length(B[-ab])
A<-length(grupos)+largo
}
}  #termina el if(length(dup)>=2)
largo<-length(B[-ab])
A<-length(grupos)+largo
} #termina el else de  if(dim(data)[1]==A)
}

}else{
#terminar el if if(length(posi_calidad)==length(c(ini,di)))
z3<-setdiff(c(di,ini), posi_calidad)
for(d in 1:length(z3)){
enlace[z3[d],ini]=0
enlace[ini,z3[d]]=0

}
posicion<-which.max(enlace[ini,])
a<-enlace[ini,posicion]
di<-which(enlace[ini,]==a)
enlace[ini,posicion]=0
posicion_columna<-which.max(enlace[,posicion])
b<-enlace[posicion_columna,posicion]
enlace[ini,posicion]<-a
if(length(di)>1){
if(dim(data)[1]==A){
grupos[[1]]<-c(ini,posicion)
ab<-unlist(grupos)
largo<-length(B[-ab])
A<-length(grupos)+largo
enlace[,posicion]=0
enlace[posicion,]=0
}else{
grupos[[length(grupos)+1]]<-c(ini,posicion)
ab<-unlist(grupos)
dup<-ab[!(!duplicated(ab) & rev(!duplicated(rev(ab))))]
```

Universidad del Bío-Bío. Sistema de Bibliotecas - Chile


```
if(length(dup)>=2){
if(length(dup)==4){
ak<-which(sapply(grupos, function(e) is.element(dup[1], e)))
grupos[[ak[1]]]<-unique(unlist(grupos[ak]))
grupos[[ak[2]]]<-NULL
largo<-length(B[-ab])
A<-length(grupos)+largo
ab<-unlist(grupos)
dup<-ab[!(!duplicated(ab) & rev(!duplicated(rev(ab))))]
ak<-which(sapply(grupos, function(e) is.element(dup[1], e)))
grupos[[ak[1]]]<-unique(unlist(grupos[ak]))
grupos[[ak[2]]]<-NULL
largo<-length(B[-ab])
A<-length(grupos)+largo
}else{
az<-which(sapply(grupos, function(e) is.element(dup[1], e)))
grupos[[az[1]]]<-unique(unlist(grupos[az]))
grupos[[az[length(az)]]]<-NULL
largo<-length(B[-ab])
A<-length(grupos)+largo}
}else{
largo<-length(B[-ab])
A<-length(grupos)+largo
}
enlace[,posicion]=0
enlace[posicion,]=0
}
di2<-which(enlace[ini,]==a)
for(j in 1:length(di2)){
if(A>alpha){
posicion<-which.max(enlace[ini,])
a2<-enlace[ini,posicion]
enlace[ini,posicion]=0
posicion_columna<-which.max(enlace[,posicion])
b2<-enlace[posicion_columna,posicion]
enlace[ini,posicion]<-a2
if(a2>=b2){
enlace[,posicion]=0
enlace[posicion,]=0
grupos[[length(grupos)+1]]<-c(ini,posicion)
```

Universidad del Bío-Bío. Sistema de Bibliotecas - Chile


```
ab<-unlist(grupos)
dup<-ab[!(!duplicated(ab) & rev(!duplicated(rev(ab))))]
if(length(dup)>=2){
if(length(dup)==4){
ak<-which(sapply(grupos, function(e) is.element(dup[1], e)))
grupos[[ak[1]]]<-unique(unlist(grupos[ak]))
grupos[[ak[2]]]<-NULL
largo<-length(B[-ab])
A<-length(grupos)+largo
ab<-unlist(grupos)
dup<-ab[!(!duplicated(ab) & rev(!duplicated(rev(ab))))]
ak<-which(sapply(grupos, function(e) is.element(dup[1], e)))
grupos[[ak[1]]]<-unique(unlist(grupos[ak]))
grupos[[ak[2]]]<-NULL
largo<-length(B[-ab])
A<-length(grupos)+largo
}else{
az<-which(sapply(grupos, function(e) is.element(dup[1], e)))
grupos[[az[1]]]<-unique(unlist(grupos[az]))
grupos[[az[length(az)]]]<-NULL
largo<-length(B[-ab])
A<-length(grupos)+largo
}
} #termina el if(a>=b)
}
} #termina if(A>alpha)
} #termina el for
#aca terminar el if(length)>=2
}else{
if(dim(data)[1]==A){
grupos[[1]]<-c(ini,posicion)
ab<-unlist(grupos)
largo<-length(B[-ab])
A<-length(grupos)+largo
enlace[,posicion]=0
enlace[posicion,]=0
}else{
grupos[[length(grupos)+1]]<-c(ini,posicion)
enlace[,posicion]=0
enlace[posicion,]=0
```

Universidad del Bío-Bío. Sistema de Bibliotecas - Chile


```
ab<-unlist(grupos)
dup<-ab[!(!duplicated(ab) & rev(!duplicated(rev(ab))))]
if(length(dup)>=2){
if(length(dup)==4){
ak<-which(sapply(grupos, function(e) is.element(dup[1], e)))
grupos[[ak[1]]]<-unique(unlist(grupos[ak]))
grupos[[ak[2]]]<-NULL
largo<-length(B[-ab])
A<-length(grupos)+largo
ab<-unlist(grupos)
dup<-ab[!(!duplicated(ab) & rev(!duplicated(rev(ab))))]
ak<-which(sapply(grupos, function(e) is.element(dup[1], e)))
grupos[[ak[1]]]<-unique(unlist(grupos[ak]))
grupos[[ak[2]]]<-NULL
largo<-length(B[-ab])
A<-length(grupos)+largo
}else{
az<-which(sapply(grupos, function(e) is.element(dup[1], e)))
grupos[[az[1]]]<-unique(unlist(grupos[az]))
grupos[[az[length(az)]]]<-NULL
largo<-length(B[-ab])
A<-length(grupos)+largo
}
}  #termina el if(length(dup)>=2)
largo<-length(B[-ab])
A<-length(grupos)+largo
} #termina el else de  if(dim(data)[1]==A)
}
}
}### aca termina el else de if(length(which(posi_calidad==ini))==0)
a<-3
b<-2
}
# termina el if(a>b)
}


if(a==b){
max3<-rep(0,length(di))
posi3<-rep(0,length(di))
```

Universidad del Bío-Bío. Sistema de Bibliotecas - Chile


```
for(np in 1:length(di)){
di2<-di[np]
ah<-which(enlace[,di2]==b)
ah<-c(ah,di2)
af<-rep(0,length(ah))
for(ap in 1:length(ah)){

af[ap]<-length(which(enlace[,ah[ap]]==b))
}
max3[np]<-ah[which.max(af)]
posi3[np]<-af[which.max(af)]
}
posi4<-which.max(posi3)
ini<-max3[posi4]
posicion<-which.max(enlace[ini,])
a<-enlace[ini,posicion]
di<-which(enlace[ini,]==a)
enlace[ini,posicion]=0
posicion_columna<-which.max(enlace[,posicion])
b<-enlace[posicion_columna,posicion]
enlace[ini,posicion]<-a
af2<-c(ini,di)
combinaciones<-list()
for(ag in 1:length(af2)){
combi <- combinations(length(af2), ag, af2)
for(an in 1:dim(combi)[1]){
combinaciones[[length(combinaciones)+1]]<-combi[an,]
}
}
Calidad_c<-rep(0,length(combinaciones))
for(kp in 1:length(combinaciones)){
grupos2<-grupos
grupos2[[length(grupos2)+1]]<-combinaciones[[kp]]
ab<-unlist(grupos2)
dup<-ab[!(!duplicated(ab) & rev(!duplicated(rev(ab))))]
if(length(dup)>=2){
if(length(dup)==4){
ak<-which(sapply(grupos2, function(e) is.element(dup[1], e)))
grupos2[[ak[1]]]<-unique(unlist(grupos2[ak]))
grupos2[[ak[2]]]<-NULL
```

Universidad del Bío-Bío. Sistema de Bibliotecas - Chile


```
largo<-length(B[-ab])
A<-length(grupos2)+largo
ab<-unlist(grupos2)
dup<-ab[!(!duplicated(ab) & rev(!duplicated(rev(ab))))]
ak<-which(sapply(grupos2, function(e) is.element(dup[1], e)))
grupos2[[ak[1]]]<-unique(unlist(grupos2[ak]))
grupos2[[ak[2]]]<-NULL
largo<-length(B[-ab])
A<-length(grupos2)+largo
}else{
az<-which(sapply(grupos2, function(e) is.element(dup[1], e)))
grupos2[[az[1]]]<-unique(unlist(grupos2[az]))
grupos2[[az[length(az)]]]<-NULL
largo<-length(B[-ab])
A<-length(grupos2)+largo
}
} #termina length(dup)>=2
ab2<-unlist(grupos2)
ac2<-B[-ab2]
z2<-list()
z2<-grupos2
for(ip in 1:length(ac2)){
z2[[length(grupos2)+ip]]<-ac2[ip]
}
class_C<-numeric()
for(n in 1:length(z2)){
class_C[z2[[n]]]<-n
}
Y_c<-matrix(0,ncol=dim(data)[1],nrow=dim(data)[1])
for(i in 1:dim(data)[1]){
for(j in 1:dim(data)[1]){
Y_c[i,j]=ifelse(class_C[i]==class_C[j],1,0)
}}
Y_c1<-(1-Y_c)
C1<-(dim(data)[2]-C)
Calidad_c[kp]<-sum(C*Y_c+C1*Y_c1)
} # termina el for kp
max_calidad<-which.max(Calidad_c)
posi_calidad<-unlist(combinaciones[max_calidad])
max_valor<-Calidad_c[max_calidad]
```

Universidad del Bío-Bío. Sistema de Bibliotecas - Chile


```
if(max_calidad==1){
ax<-which(Calidad_c==max_valor)
Calidad_c[ax]=0
max_calidad<-which.max(Calidad_c)
posi_calidad<-unlist(combinaciones[max_calidad])
max_valor<-Calidad_c[max_calidad]
}
freq<-which(Calidad_c==max_valor)
ae<-rep(0,length(freq))
for(ld in 1:length(freq)){
ae[ld]<- length(combinaciones[[freq[ld]]])
}
ae2<-which(ae==length(posi_calidad))
freq<-freq[ae2]
if(length(freq)!=1){
z3<-combinaciones[freq]
opc<-vector(mode = "list", length = length(z3))
grupos_aux<-grupos
for(kl in 1:length(z3)){
grupos_aux<-grupos
grupos2<-grupos_aux
grupos2[[length(grupos2)+1]]<-unlist(z3[[kl]])
enlace2<-enlace
ceros_p<-length(unlist(z3[[kl]]))
ceros<-unlist(z3[[kl]][2:ceros_p])
enlace2[ceros,]=0
enlace2[,ceros]=0
grupos_aux<-grupos2
ab<-unlist(grupos_aux)
dup<-ab[!(!duplicated(ab) & rev(!duplicated(rev(ab))))]
if(length(dup)>=2){
if(length(dup)==4){
ak<-which(sapply(grupos2, function(e) is.element(dup[1], e)))
grupos_aux[[ak[1]]]<-unique(unlist(grupos_aux[ak]))
grupos_aux[[ak[2]]]<-NULL
largo<-length(B[-ab])
A<-length(grupos_aux)+largo
ab<-unlist(grupos_aux)
dup<-ab[!(!duplicated(ab) & rev(!duplicated(rev(ab))))]
ak<-which(sapply(grupos_aux, function(e) is.element(dup[1], e)))
```

Universidad del Bío-Bío. Sistema de Bibliotecas - Chile


```
grupos_aux[[ak[1]]]<-unique(unlist(grupos_aux[ak]))
grupos_aux[[ak[2]]]<-NULL
largo<-length(B[-ab])
A<-length(grupos_aux)+largo
}else{
az<-which(sapply(grupos_aux, function(e) is.element(dup[1], e)))
grupos_aux[[az[1]]]<-unique(unlist(grupos_aux[az]))
grupos_aux[[az[length(az)]]]<-NULL
largo<-length(B[-ab])
A<-length(grupos_aux)+largo
}
} #termina length(dup)>=2
max_calidad2<-2
for(kl2 in 1:3){
grupos2<-grupos_aux
if(max_calidad2==1){
k<-numeric(dim(data)[1])
for(j in 1:dim(data)[1]){
k[j]<-max(enlace2[,j])
}
ab<-unlist(grupos2)
#tengo que cambiar ese A por un B
if(A==dim(data)[1]){
ini<-which.max(k)
}else{
k[ab]=0
k[c(ini,di)]=0
ini<-which.max(k)
}
suma3<-sum(k)
if(suma3==0){
opc[[kl]][[kl2]]=0
max_calidad2<-1
}else{
posicion<-which.max(enlace2[ini,])
a<-enlace2[ini,posicion]
di<-which(enlace2[ini,]==a)
enlace2[ini,posicion]=0
posicion_columna<-which.max(enlace2[,posicion])
b<-enlace2[posicion_columna,posicion]
```

Universidad del Bío-Bío. Sistema de Bibliotecas - Chile


```
enlace2[ini,posicion]<-a
max3<-rep(0,length(di))
posi3<-rep(0,length(di))
for(np in 1:length(di)){
di2<-di[np]
ah<-which(enlace2[,di2]==b)
ah<-c(ah,di2)
af<-rep(0,length(ah))
for(ap in 1:length(ah)){
af[ap]<-length(which(enlace2[,ah[ap]]==b))
}
max3[np]<-ah[which.max(af)]
posi3[np]<-af[which.max(af)]
}
posi4<-which.max(posi3)
ini<-max3[posi4]
posicion<-which.max(enlace2[ini,])
a<-enlace2[ini,posicion]
di<-which(enlace2[ini,]==a)
enlace2[ini,posicion]=0
posicion_columna<-which.max(enlace2[,posicion])
b<-enlace2[posicion_columna,posicion]
enlace2[ini,posicion]<-a
af2<-c(ini,di)
combinaciones<-list()
for(ag in 1:length(af2)){
combi <- combinations(length(af2), ag, af2)
for(an in 1:dim(combi)[1]){
combinaciones[[length(combinaciones)+1]]<-combi[an,]
}
}
Calidad_c<-rep(0,length(combinaciones))
for(kp in 1:length(combinaciones)){
grupos2<-grupos_aux
grupos2[[length(grupos2)+1]]<-combinaciones[[kp]]
ab<-unlist(grupos2)
dup<-ab[!(!duplicated(ab) & rev(!duplicated(rev(ab))))]
if(length(dup)>=2){
if(length(dup)==4){
ak<-which(sapply(grupos2, function(e) is.element(dup[1], e)))
```

Universidad del Bío-Bío. Sistema de Bibliotecas - Chile


```
grupos2[[ak[1]]]<-unique(unlist(grupos2[ak]))
grupos2[[ak[2]]]<-NULL
largo<-length(B[-ab])
A<-length(grupos2)+largo
ab<-unlist(grupos2)
dup<-ab[!(!duplicated(ab) & rev(!duplicated(rev(ab))))]
ak<-which(sapply(grupos2, function(e) is.element(dup[1], e)))
grupos2[[ak[1]]]<-unique(unlist(grupos2[ak]))
grupos2[[ak[2]]]<-NULL
largo<-length(B[-ab])
A<-length(grupos2)+largo
}else{
az<-which(sapply(grupos2, function(e) is.element(dup[1], e)))
grupos2[[az[1]]]<-unique(unlist(grupos2[az]))
grupos2[[az[length(az)]]]<-NULL
largo<-length(B[-ab])
A<-length(grupos2)+largo
}
} #termina length(dup)>=2
ab2<-unlist(grupos2)
ac2<-B[-ab2]
z2<-list()
z2<-grupos2
for(ip in 1:length(ac2)){
z2[[length(grupos2)+ip]]<-ac2[ip]
}
class_C<-numeric()
for(n in 1:length(z2)){
class_C[z2[[n]]]<-n
}
Y_c<-matrix(0,ncol=dim(data)[1],nrow=dim(data)[1])
for(i in 1:dim(data)[1]){
for(j in 1:dim(data)[1]){
Y_c[i,j]=ifelse(class_C[i]==class_C[j],1,0)
}}
Y_c1<-(1-Y_c)
C1<-(dim(data)[2]-C)
Calidad_c[kp]<-sum(C*Y_c+C1*Y_c1)
} # termina el for kp
max_calidad<-which.max(Calidad_c)
```

Universidad del Bío-Bío. Sistema de Bibliotecas - Chile


```
posi_calidad<-unlist(combinaciones[max_calidad])
max_valor<-Calidad_c[max_calidad]
if(max_calidad!=1){
grupos_aux[[length(grupos_aux)+1]]<-combinaciones[[max_calidad]]
ab<-unlist(grupos_aux)
dup<-ab[!(!duplicated(ab) & rev(!duplicated(rev(ab))))]
if(length(dup)>=2){
if(length(dup)==4){
ak<-which(sapply(grupos_aux, function(e) is.element(dup[1], e)))
grupos_aux[[ak[1]]]<-unique(unlist(grupos_aux[ak]))
grupos_aux[[ak[2]]]<-NULL
largo<-length(B[-ab])
A<-length(grupos_aux)+largo
ab<-unlist(grupos_aux)
dup<-ab[!(!duplicated(ab) & rev(!duplicated(rev(ab))))]
ak<-which(sapply(grupos_aux, function(e) is.element(dup[1], e)))
grupos_aux[[ak[1]]]<-unique(unlist(grupos_aux[ak]))
grupos_aux[[ak[2]]]<-NULL
largo<-length(B[-ab])
A<-length(grupos_aux)+largo
}else{
az<-which(sapply(grupos_aux, function(e) is.element(dup[1], e)))
grupos_aux[[az[1]]]<-unique(unlist(grupos_aux[az]))
grupos_aux[[az[length(az)]]]<-NULL
largo<-length(B[-ab])
A<-length(grupos_aux)+largo
}
} #termina length(dup)>=2
opc[[kl]][kl2]=max_valor
ceros_p<-which(posi_calidad!=ini)
ceros<-posi_calidad[ceros_p]
enlace2[ceros,]=0
enlace2[,ceros]=0
max_calidad2<-max_calidad
#termina if(max_calidad!=1)
}else{
opc[[kl]][kl2]=max_valor
max_calidad2<-max_calidad
}
} ####termina el else de if(sum3==0)
```

Universidad del Bío-Bío. Sistema de Bibliotecas - Chile


```
##### termina el if(max_calidad2)
}else{
k<-numeric(dim(data)[1])
for(j in 1:dim(data)[1]){
k[j]<-max(enlace2[,j])
}
ab<-unlist(grupos2)
if(A==dim(data)[1]){
ini<-which.max(k)
}else{
k[ab]=0
ini<-which.max(k)
}
posicion<-which.max(enlace2[ini,])
a<-enlace2[ini,posicion]
di<-which(enlace2[ini,]==a)
enlace2[ini,posicion]=0
posicion_columna<-which.max(enlace2[,posicion])
b<-enlace2[posicion_columna,posicion]
enlace2[ini,posicion]<-a
max3<-rep(0,length(di))
posi3<-rep(0,length(di))
for(np in 1:length(di)){
di2<-di[np]
ah<-which(enlace2[,di2]==b)
ah<-c(ah,di2)
af<-rep(0,length(ah))
for(ap in 1:length(ah)){
af[ap]<-length(which(enlace2[,ah[ap]]==b))
}
max3[np]<-ah[which.max(af)]
posi3[np]<-af[which.max(af)]
}
posi4<-which.max(posi3)
ini<-max3[posi4]
posicion<-which.max(enlace2[ini,])
a<-enlace2[ini,posicion]
di<-which(enlace2[ini,]==a)
enlace2[ini,posicion]=0
posicion_columna<-which.max(enlace2[,posicion])
```

Universidad del Bío-Bío. Sistema de Bibliotecas - Chile


```
b<-enlace2[posicion_columna,posicion]
enlace2[ini,posicion]<-a
af2<-c(ini,di)
combinaciones<-list()
for(ag in 1:length(af2)){
combi <- combinations(length(af2), ag, af2)
for(an in 1:dim(combi)[1]){
combinaciones[[length(combinaciones)+1]]<-combi[an,]
}
}
Calidad_c<-rep(0,length(combinaciones))
for(kp in 1:length(combinaciones)){
grupos2<-grupos_aux
grupos2[[length(grupos2)+1]]<-combinaciones[[kp]]
ab<-unlist(grupos2)
dup<-ab[!(!duplicated(ab) & rev(!duplicated(rev(ab))))]
if(length(dup)>=2){
if(length(dup)==4){
ak<-which(sapply(grupos2, function(e) is.element(dup[1], e)))
grupos2[[ak[1]]]<-unique(unlist(grupos2[ak]))
grupos2[[ak[2]]]<-NULL
largo<-length(B[-ab])
A<-length(grupos2)+largo
ab<-unlist(grupos2)
dup<-ab[!(!duplicated(ab) & rev(!duplicated(rev(ab))))]
ak<-which(sapply(grupos2, function(e) is.element(dup[1], e)))
grupos2[[ak[1]]]<-unique(unlist(grupos2[ak]))
grupos2[[ak[2]]]<-NULL
largo<-length(B[-ab])
A<-length(grupos2)+largo
}else{
az<-which(sapply(grupos2, function(e) is.element(dup[1], e)))
grupos2[[az[1]]]<-unique(unlist(grupos2[az]))
grupos2[[az[length(az)]]]<-NULL
largo<-length(B[-ab])
A<-length(grupos2)+largo
}
} #termina length(dup)>=2
ab2<-unlist(grupos2)
ac2<-B[-ab2]
```

Universidad del Bío-Bío. Sistema de Bibliotecas - Chile


```
z2<-list()
z2<-grupos2
for(ip in 1:length(ac2)){
z2[[length(grupos2)+ip]]<-ac2[ip]
}
class_C<-numeric()
for(n in 1:length(z2)){
class_C[z2[[n]]]<-n
}
Y_c<-matrix(0,ncol=dim(data)[1],nrow=dim(data)[1])
for(i in 1:dim(data)[1]){
for(j in 1:dim(data)[1]){
Y_c[i,j]=ifelse(class_C[i]==class_C[j],1,0)
}}
Y_c1<-(1-Y_c)
C1<-(dim(data)[2]-C)
Calidad_c[kp]<-sum(C*Y_c+C1*Y_c1)
} # termina el for kp
max_calidad<-which.max(Calidad_c)
posi_calidad<-unlist(combinaciones[max_calidad])
max_valor<-Calidad_c[max_calidad]
if(max_calidad!=1){
grupos_aux[[length(grupos_aux)+1]]<-combinaciones[[max_calidad]]
ab<-unlist(grupos_aux)
dup<-ab[!(!duplicated(ab) & rev(!duplicated(rev(ab))))]
if(length(dup)>=2){
if(length(dup)==4){
ak<-which(sapply(grupos_aux, function(e) is.element(dup[1], e)))
grupos_aux[[ak[1]]]<-unique(unlist(grupos_aux[ak]))
grupos_aux[[ak[2]]]<-NULL
largo<-length(B[-ab])
A<-length(grupos_aux)+largo
ab<-unlist(grupos_aux)
dup<-ab[!(!duplicated(ab) & rev(!duplicated(rev(ab))))]
ak<-which(sapply(grupos_aux, function(e) is.element(dup[1], e)))
grupos_aux[[ak[1]]]<-unique(unlist(grupos_aux[ak]))
grupos_aux[[ak[2]]]<-NULL
largo<-length(B[-ab])
A<-length(grupos_aux)+largo
}else{
```

Universidad del Bío-Bío. Sistema de Bibliotecas - Chile


```
az<-which(sapply(grupos_aux, function(e) is.element(dup[1], e)))
grupos_aux[[az[1]]]<-unique(unlist(grupos_aux[az]))
grupos_aux[[az[length(az)]]]<-NULL
largo<-length(B[-ab])
A<-length(grupos_aux)+largo
}
} #termina length(dup)>=2
opc[[kl]][kl2]=max_valor
ceros_p<-which(posi_calidad!=ini)
ceros<-posi_calidad[ceros_p]
enlace2[ceros,]=0
enlace2[,ceros]=0
max_calidad2<-max_calidad
##termina el if(max_calidad!=1)
}else{
opc[[kl]][kl2]=max_valor
max_calidad2<-max_calidad
}
} # termina el else de if(max_calidad2)
}######### termina el for(kl2 in 1:3)

}########## termina el for(kl in 1:length(z3))
ke<-rep(0,length(z3))
for(kw in 1:length(z3)){
ke[kw]<-opc[[kw]][which.max(opc[[kw]])]
}
maximo<-which.max(ke)
k<-numeric(dim(data)[1])
for(j in 1:dim(data)[1]){
k[j]<-max(enlace[,j])
}
ab<-unlist(grupos)
#tengo que cambiar ese A por un B
if(A==dim(data)[1]){
ini<-which.max(k)
}else{
k[ab]=0
ini<-which.max(k)
}
posicion<-which.max(enlace[ini,])
```

Universidad del Bío-Bío. Sistema de Bibliotecas - Chile


```
a<-enlace[ini,posicion]
di<-which(enlace[ini,]==a)
enlace[ini,posicion]=0
posicion_columna<-which.max(enlace[,posicion])
b<-enlace[posicion_columna,posicion]
enlace[ini,posicion]<-a
max3<-rep(0,length(di))
posi3<-rep(0,length(di))
for(np in 1:length(di)){
di2<-di[np]
ah<-which(enlace[,di2]==b)
ah<-c(ah,di2)
af<-rep(0,length(ah))
for(ap in 1:length(ah)){
af[ap]<-length(which(enlace[,ah[ap]]==b))
}
max3[np]<-ah[which.max(af)]
posi3[np]<-af[which.max(af)]
}
posi4<-which.max(posi3)
ini<-max3[posi4]
posicion<-which.max(enlace[ini,])
a<-enlace[ini,posicion]
di<-which(enlace[ini,]==a)
enlace[ini,posicion]=0
posicion_columna<-which.max(enlace[,posicion])
b<-enlace[posicion_columna,posicion]
enlace[ini,posicion]<-a
posi_calidad<-z3[[maximo]]
}  # termina el if(length(freq)!=1)
if(length(which(posi_calidad==ini))==0){
ini<-posi_calidad[1]
di<-posi_calidad[-1]
grupos[[length(grupos)+1]]<-c(ini,di[1])
ab<-unlist(grupos)
dup<-ab[!(!duplicated(ab) & rev(!duplicated(rev(ab))))]
if(length(dup)>=2){
if(length(dup)==4){
ak<-which(sapply(grupos, function(e) is.element(dup[1], e)))
grupos[[ak[1]]]<-unique(unlist(grupos[ak]))
```

Universidad del Bío-Bío. Sistema de Bibliotecas - Chile


```
grupos[[ak[2]]]<-NULL
largo<-length(B[-ab])
A<-length(grupos)+largo
ab<-unlist(grupos)
dup<-ab[!(!duplicated(ab) & rev(!duplicated(rev(ab))))]
ak<-which(sapply(grupos, function(e) is.element(dup[1], e)))
grupos[[ak[1]]]<-unique(unlist(grupos[ak]))
grupos[[ak[2]]]<-NULL
largo<-length(B[-ab])
A<-length(grupos)+largo
}else{
az<-which(sapply(grupos, function(e) is.element(dup[1], e)))
grupos[[az[1]]]<-unique(unlist(grupos[az]))
grupos[[az[length(az)]]]<-NULL
largo<-length(B[-ab])
A<-length(grupos)+largo}
}else{
largo<-length(B[-ab])
A<-length(grupos)+largo
}
enlace[,di[1]]=0
enlace[di[1],]=0
di<-di[-1]
if(length(di)!=0){
for(j in 1:length(di)){
if(A>alpha){
enlace[,di[j]]=0
enlace[di[j],]=0
grupos[[length(grupos)+1]]<-c(ini,di[j])
ab<-unlist(grupos)
dup<-ab[!(!duplicated(ab) & rev(!duplicated(rev(ab))))]
if(length(dup)>=2){
if(length(dup)==4){
ak<-which(sapply(grupos, function(e) is.element(dup[1], e)))
grupos[[ak[1]]]<-unique(unlist(grupos[ak]))
grupos[[ak[2]]]<-NULL
largo<-length(B[-ab])
A<-length(grupos)+largo
ab<-unlist(grupos)
dup<-ab[!(!duplicated(ab) & rev(!duplicated(rev(ab))))]
```

Universidad del Bío-Bío. Sistema de Bibliotecas - Chile


```
ak<-which(sapply(grupos, function(e) is.element(dup[1], e)))
grupos[[ak[1]]]<-unique(unlist(grupos[ak]))
grupos[[ak[2]]]<-NULL
largo<-length(B[-ab])
A<-length(grupos)+largo
}else{
az<-which(sapply(grupos, function(e) is.element(dup[1], e)))
grupos[[az[1]]]<-unique(unlist(grupos[az]))
grupos[[az[length(az)]]]<-NULL
largo<-length(B[-ab])
A<-length(grupos)+largo
}
} ####termina el if(length(dup)>=2)
} #termina if(A>alpha)
} #termina el for
} ##termina el length(di)!=0
posi_calidad<-numeric
}else{ ### el else de if(length(which(posi_calidad==ini))==0)
if(length(posi_calidad)==length(c(ini,di))){
if(length(di)>1){
if(dim(data)[1]==A){
grupos[[1]]<-c(ini,posicion)
ab<-unlist(grupos)
largo<-length(B[-ab])
A<-length(grupos)+largo
enlace[,posicion]=0
enlace[posicion,]=0
}else{
grupos[[length(grupos)+1]]<-c(ini,posicion)
ab<-unlist(grupos)
dup<-ab[!(!duplicated(ab) & rev(!duplicated(rev(ab))))]
if(length(dup)>=2){
if(length(dup)==4){
ak<-which(sapply(grupos, function(e) is.element(dup[1], e)))
grupos[[ak[1]]]<-unique(unlist(grupos[ak]))
grupos[[ak[2]]]<-NULL
largo<-length(B[-ab])
A<-length(grupos)+largo
ab<-unlist(grupos)
dup<-ab[!(!duplicated(ab) & rev(!duplicated(rev(ab))))]
```

Universidad del Bío-Bío. Sistema de Bibliotecas - Chile


```
ak<-which(sapply(grupos, function(e) is.element(dup[1], e)))
grupos[[ak[1]]]<-unique(unlist(grupos[ak]))
grupos[[ak[2]]]<-NULL
largo<-length(B[-ab])
A<-length(grupos)+largo
}else{
az<-which(sapply(grupos, function(e) is.element(dup[1], e)))
grupos[[az[1]]]<-unique(unlist(grupos[az]))
grupos[[az[length(az)]]]<-NULL
largo<-length(B[-ab])
A<-length(grupos)+largo}
}else{
largo<-length(B[-ab])
A<-length(grupos)+largo
}
enlace[,posicion]=0
enlace[posicion,]=0
}
di2<-which(enlace[ini,]==a)
for(j in 1:length(di2)){
if(A>alpha){
posicion<-which.max(enlace[ini,])
a2<-enlace[ini,posicion]
enlace[ini,posicion]=0
posicion_columna<-which.max(enlace[,posicion])
b2<-enlace[posicion_columna,posicion]
enlace[ini,posicion]<-a2
if(a2>=b2){
enlace[,posicion]=0
enlace[posicion,]=0
grupos[[length(grupos)+1]]<-c(ini,posicion)
ab<-unlist(grupos)
dup<-ab[!(!duplicated(ab) & rev(!duplicated(rev(ab))))]
if(length(dup)>=2){
if(length(dup)==4){
ak<-which(sapply(grupos, function(e) is.element(dup[1], e)))
grupos[[ak[1]]]<-unique(unlist(grupos[ak]))
grupos[[ak[2]]]<-NULL
largo<-length(B[-ab])
A<-length(grupos)+largo
```

Universidad del Bío-Bío. Sistema de Bibliotecas - Chile


```
ab<-unlist(grupos)
dup<-ab[!(!duplicated(ab) & rev(!duplicated(rev(ab))))]
ak<-which(sapply(grupos, function(e) is.element(dup[1], e)))
grupos[[ak[1]]]<-unique(unlist(grupos[ak]))
grupos[[ak[2]]]<-NULL
largo<-length(B[-ab])
A<-length(grupos)+largo
}else{
az<-which(sapply(grupos, function(e) is.element(dup[1], e)))
grupos[[az[1]]]<-unique(unlist(grupos[az]))
grupos[[az[length(az)]]]<-NULL
largo<-length(B[-ab])
A<-length(grupos)+largo
}
} #termina el if(a>=b)
}
} #termina if(A>alpha)
} #termina el for
#aca terminar el if(length)>=2
}else{
if(dim(data)[1]==A){
grupos[[1]]<-c(ini,posicion)
ab<-unlist(grupos)
largo<-length(B[-ab])
A<-length(grupos)+largo
enlace[,posicion]=0
enlace[posicion,]=0
}else{
grupos[[length(grupos)+1]]<-c(ini,posicion)
enlace[,posicion]=0
enlace[posicion,]=0
ab<-unlist(grupos)
dup<-ab[!(!duplicated(ab) & rev(!duplicated(rev(ab))))]
if(length(dup)>=2){
if(length(dup)==4){
ak<-which(sapply(grupos, function(e) is.element(dup[1], e)))
grupos[[ak[1]]]<-unique(unlist(grupos[ak]))
grupos[[ak[2]]]<-NULL
largo<-length(B[-ab])
A<-length(grupos)+largo
```

Universidad del Bío-Bío. Sistema de Bibliotecas - Chile


```
ab<-unlist(grupos)
dup<-ab[!(!duplicated(ab) & rev(!duplicated(rev(ab))))]
ak<-which(sapply(grupos, function(e) is.element(dup[1], e)))
grupos[[ak[1]]]<-unique(unlist(grupos[ak]))
grupos[[ak[2]]]<-NULL
largo<-length(B[-ab])
A<-length(grupos)+largo
}else{
az<-which(sapply(grupos, function(e) is.element(dup[1], e)))
grupos[[az[1]]]<-unique(unlist(grupos[az]))
grupos[[az[length(az)]]]<-NULL
largo<-length(B[-ab])
A<-length(grupos)+largo
}
}  #termina el if(length(dup)>=2)
largo<-length(B[-ab])
A<-length(grupos)+largo
} #termina el else de  if(dim(data)[1]==A)
}
}else{
#terminar el if if(length(posi_calidad)==length(c(ini,di)))
z3<-setdiff(c(di,ini), posi_calidad)
for(d in 1:length(z3)){
enlace[z3[d],ini]=0
enlace[ini,z3[d]]=0
}
posicion<-which.max(enlace[ini,])
a<-enlace[ini,posicion]
di<-which(enlace[ini,]==a)
enlace[ini,posicion]=0
posicion_columna<-which.max(enlace[,posicion])
b<-enlace[posicion_columna,posicion]
enlace[ini,posicion]<-a
if(length(di)>1){
if(dim(data)[1]==A){
grupos[[1]]<-c(ini,posicion)
ab<-unlist(grupos)
largo<-length(B[-ab])
A<-length(grupos)+largo
enlace[,posicion]=0
```

Universidad del Bío-Bío. Sistema de Bibliotecas - Chile


```
enlace[posicion,]=0
}else{
grupos[[length(grupos)+1]]<-c(ini,posicion)
ab<-unlist(grupos)
dup<-ab[!(!duplicated(ab) & rev(!duplicated(rev(ab))))]
if(length(dup)>=2){
if(length(dup)==4){
ak<-which(sapply(grupos, function(e) is.element(dup[1], e)))
grupos[[ak[1]]]<-unique(unlist(grupos[ak]))
grupos[[ak[2]]]<-NULL
largo<-length(B[-ab])
A<-length(grupos)+largo
ab<-unlist(grupos)
dup<-ab[!(!duplicated(ab) & rev(!duplicated(rev(ab))))]
ak<-which(sapply(grupos, function(e) is.element(dup[1], e)))
grupos[[ak[1]]]<-unique(unlist(grupos[ak]))
grupos[[ak[2]]]<-NULL
largo<-length(B[-ab])
A<-length(grupos)+largo
}else{
az<-which(sapply(grupos, function(e) is.element(dup[1], e)))
grupos[[az[1]]]<-unique(unlist(grupos[az]))
grupos[[az[length(az)]]]<-NULL
largo<-length(B[-ab])
A<-length(grupos)+largo}
}else{
largo<-length(B[-ab])
A<-length(grupos)+largo
}
enlace[,posicion]=0
enlace[posicion,]=0
}
di2<-which(enlace[ini,]==a)
for(j in 1:length(di2)){
if(A>alpha){
posicion<-which.max(enlace[ini,])
a2<-enlace[ini,posicion]
enlace[ini,posicion]=0
posicion_columna<-which.max(enlace[,posicion])
b2<-enlace[posicion_columna,posicion]
```

Universidad del Bío-Bío. Sistema de Bibliotecas - Chile


```
enlace[ini,posicion]<-a2
if(a2>=b2){
enlace[,posicion]=0
enlace[posicion,]=0
grupos[[length(grupos)+1]]<-c(ini,posicion)
ab<-unlist(grupos)
dup<-ab[!(!duplicated(ab) & rev(!duplicated(rev(ab))))]
if(length(dup)>=2){
if(length(dup)==4){
ak<-which(sapply(grupos, function(e) is.element(dup[1], e)))
grupos[[ak[1]]]<-unique(unlist(grupos[ak]))
grupos[[ak[2]]]<-NULL
largo<-length(B[-ab])
A<-length(grupos)+largo
ab<-unlist(grupos)
dup<-ab[!(!duplicated(ab) & rev(!duplicated(rev(ab))))]
ak<-which(sapply(grupos, function(e) is.element(dup[1], e)))
grupos[[ak[1]]]<-unique(unlist(grupos[ak]))
grupos[[ak[2]]]<-NULL
largo<-length(B[-ab])
A<-length(grupos)+largo
}else{
az<-which(sapply(grupos, function(e) is.element(dup[1], e)))
grupos[[az[1]]]<-unique(unlist(grupos[az]))
grupos[[az[length(az)]]]<-NULL
largo<-length(B[-ab])
A<-length(grupos)+largo
}
} #termina el if(a>=b)
}
} #termina if(A>alpha)
} #termina el for
#aca terminar el if(length)>=2
}else{
if(dim(data)[1]==A){
grupos[[1]]<-c(ini,posicion)
ab<-unlist(grupos)
largo<-length(B[-ab])
A<-length(grupos)+largo
enlace[,posicion]=0
```

Universidad del Bío-Bío. Sistema de Bibliotecas - Chile


```
enlace[posicion,]=0
}else{
grupos[[length(grupos)+1]]<-c(ini,posicion)
enlace[,posicion]=0
enlace[posicion,]=0
ab<-unlist(grupos)
dup<-ab[!(!duplicated(ab) & rev(!duplicated(rev(ab))))]
if(length(dup)>=2){
if(length(dup)==4){
ak<-which(sapply(grupos, function(e) is.element(dup[1], e)))
grupos[[ak[1]]]<-unique(unlist(grupos[ak]))
grupos[[ak[2]]]<-NULL
largo<-length(B[-ab])
A<-length(grupos)+largo
ab<-unlist(grupos)
dup<-ab[!(!duplicated(ab) & rev(!duplicated(rev(ab))))]
ak<-which(sapply(grupos, function(e) is.element(dup[1], e)))
grupos[[ak[1]]]<-unique(unlist(grupos[ak]))
grupos[[ak[2]]]<-NULL
largo<-length(B[-ab])
A<-length(grupos)+largo
}else{
az<-which(sapply(grupos, function(e) is.element(dup[1], e)))
grupos[[az[1]]]<-unique(unlist(grupos[az]))
grupos[[az[length(az)]]]<-NULL
largo<-length(B[-ab])
A<-length(grupos)+largo
}
}  #termina el if(length(dup)>=2)
largo<-length(B[-ab])
A<-length(grupos)+largo
} #termina el else de  if(dim(data)[1]==A)
}
}
}### aca termina el else de if(length(which(posi_calidad==ini))==0)
}
# termina el if(a==b)

k<-numeric(dim(data)[1])
for(j in 1:dim(data)[1]){
```

Universidad del Bío-Bío. Sistema de Bibliotecas - Chile


```
k[j]<-max(enlace[,j])
}
ab<-unlist(grupos)
k[ab]=0
largo<-length(B[-ab])
if(sum(k)==0 || largo==0){
suma<-0
}else{
suma<-sum(k)
}
k2<-length(k[k!=0])
if(k2==1){
suma<-0
}
} # Termina el while

while(A>alpha && suma==0){
ab3<-length(unlist(grupos))
if(ab3==dim(data)[1]){
af2<-1:length(grupos)
combinaciones<-list()
combi <- combinations(length(af2), 2, af2)
for(an in 1:dim(combi)[1]){
combinaciones[[an]]<-combi[an,]
}
Calidad_c<-rep(0,length(combinaciones))
for(kp in 1:length(Calidad_c)){
grupos2<-grupos
grupos2[[length(grupos2)+1]]<-unlist(grupos2[combinaciones[[kp]]])
grupos2[combinaciones[[kp]]]<-NULL
class_C<-numeric()
for(n in 1:length(grupos2)){
class_C[grupos2[[n]]]<-n
}
Y_c<-matrix(0,ncol=dim(data)[1],nrow=dim(data)[1])
for(i in 1:dim(data)[1]){
for(j in 1:dim(data)[1]){
Y_c[i,j]=ifelse(class_C[i]==class_C[j],1,0)
}}
Y_c1<-(1-Y_c)
```

Universidad del Bío-Bío. Sistema de Bibliotecas - Chile


```
C1<-(dim(data)[2]-C)
Calidad_c[kp]<-sum(C*Y_c+C1*Y_c1)
} # termina el for kp
max_calidad<-which.max(Calidad_c)
posi_calidad<-unlist(combinaciones[max_calidad])
grupos[[posi_calidad[1]]]<-unlist(grupos[posi_calidad])
grupos[posi_calidad[2]]<-NULL
ab<-unlist(grupos)
largo<-length(B[-ab])
A<-length(grupos)+largo
}else{
ab<-unlist(grupos)
largo2<-B[-ab]
grupos[[length(grupos)+1]]<-largo2
af2<-1:length(grupos)
combinaciones<-list()
combi <- combinations(length(af2), 2, af2)
for(an in 1:dim(combi)[1]){
combinaciones[[an]]<-combi[an,]
}
Calidad_c<-rep(0,length(combinaciones))
for(kp in 1:length(Calidad_c)){
grupos2<-grupos
grupos2[[length(grupos2)+1]]<-unlist(grupos2[combinaciones[[kp]]])
grupos2[combinaciones[[kp]]]<-NULL
class_C<-numeric()
for(n in 1:length(grupos2)){
class_C[grupos2[[n]]]<-n
}
Y_c<-matrix(0,ncol=dim(data)[1],nrow=dim(data)[1])
for(i in 1:dim(data)[1]){
for(j in 1:dim(data)[1]){
Y_c[i,j]=ifelse(class_C[i]==class_C[j],1,0)
}}
Y_c1<-(1-Y_c)
C1<-(dim(data)[2]-C)
Calidad_c[kp]<-sum(C*Y_c+C1*Y_c1)
} # termina el for kp
max_calidad<-which.max(Calidad_c)
posi_calidad<-unlist(combinaciones[max_calidad])
```

Universidad del Bío-Bío. Sistema de Bibliotecas - Chile


```
grupos[[posi_calidad[1]]]<-unlist(grupos[posi_calidad])
grupos[posi_calidad[2]]<-NULL
ab<-unlist(grupos)
largo<-length(B[-ab])
A<-length(grupos)+largo
}#termina el else de if(ab3==dim(data)[1]).
} #termina el while
z<-list()
z<-grupos
ab<-unlist(grupos)
ac<-B[-ab]
if(length(grupos)==alpha){
z<-grupos
}else{
for(i in 1:length(ac)){
z[[length(grupos)+i]]<-ac[i]
}
}
return(z)
}
```