UNIVERSITAT JAUME I

Grado en Matemática Computacional

Trabajo Final de Grado

Análisis cluster de variables: un puente de software para relacionar el enfoque del ASI con el enfoque clásico

Autor:
Feng Carlos LIN LIN

Tutor académico:
Pablo GREGORI HUERTA

Fecha de lectura: __ de Junio de 2022
Curso académico 2021/2022

Resumen

En este trabajo partimos de los conocimientos adquiridos en la asignatura de *Fundamentos estadísticos de la minería de datos* para presentar distintos tipos de análisis clúster y la realización de una mejora en el paquete *rchic* de R. Comenzaremos introduciendo el concepto de clúster para poder presentar y comparar el análisis clúster estándar con el análisis clúster ASI y finalmente veremos la mejora relizada que nos permite hacer compatibles los árboles sacados del ASI con la clase *hclust* y así poder hacer uso de las funciones que brindan a los objetos de esta clase.

Palabras clave

Clúster, dendrograma, rchic, similitud

Keywords

Cluster, dendrogram, rchic, similarity

Índice general

Capítulo 1

Introducción

1.1. Contexto y motivación del proyecto

Este proyecto abarca la realización del Trabajo de Fin de Grado en el Grado de Matemática Computacional.

Durante la realización de la asignatura de Minería de datos, aprendí lo que era el análisis clúster e hice un proyecto final de la asignatura que consistía en realizar un análisis clúster sobre una base de datos con las estadísticas de futbolistas de la Liga BBVA. La satisfacción al realizar este trabajo me sirvió como punto de partida para mi entusiasmo y curiosidad sobre el análisis clúster. Por eso decidí aceptar hacer estre trabajo de final de grado.

Capítulo 2

Motivación y Objetivos

Gracias a la tecnología que disponemos hoy en día, cada vez tenemos más y más datos. Este hecho en cierto modo nos obliga a desarrollar técnicas para poder analizarlos e interpretar dichos análisis.

El objetivo de este proyecto es el de introducir el análisis clúster y comparar distintos tipos de análisis, además de realizar una mejora en el paquete *rchic* en R para que los árboles, generados en la función *rchic*, puedan ser almacenados como objetos de la clase *hclust* y así poder utilizar las funciones que otorga esta clase y entonces facilitar el trabajo a investigadores que trabajen con el paquete *rchic* ya que por el momento tienen que crear a mano el objeto *hclust* con los datos del árbol creado por la función *rchic*.

Capítulo 3

Desarrollo del TFG

3.1. Análisis clúster

3.1.1. Introducción

El análisis clúster es una técnica de análisis de datos cuyo objetivo es el de agrupar cada uno de los elementos de un conjunto de datos en distintos grupos, de manera que los elementos que estén en el mismo grupo sean muy similares entre ellos y muy distintos respecto con elementos de otros grupos. Estos grupos son los llamados clústers. El grado de similaridad entre dos elementos dependerá de los valores adoptados sobre el conjunto de variables, donde se utilizará una medida de proximidad o distancia, de entre muchas posibles, para entonces poder hacer la clasificación.

Los conjuntos de datos con los que se trabaja son conjuntos de elementos multivariantes, es decir, como se puede ver en la matriz (3.1), nuestras muestras serán matrices $n \times p$, con n observaciones y p variables. x_i hace referencia al elemento i-ésimo y x_{ij} es el valor de la variable j-ésima del elemento i-ésimo.

$$X = \begin{bmatrix} x_{11} & x_{12} & \dots & x_{1p} \\ x_{21} & x_{22} & \dots & x_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ x_{n1} & x_{n2} & \dots & x_{np} \end{bmatrix} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} \tag{3.1}$$

Antes de realizar el análisis es importante plantear si partir de varios grupos ya establecidos o si considerar cada elemento como un clúster y posteriormente ir agrupándolos hasta llegar a los clústers finales.

El análisis constará de un algoritmo de clasificación del que se obtendrán las particiones, con el criterio establecido. Existen numerosos procedimientos para lograr el objetivo, principalmente destacaremos dos:

- **Métodos de partición**: Agrupan los elementos en un número de grupos previamente elegido.
- **Métodos jerárquicos**: Se construye toda una jerarquía y a partir de ella se deciden los grupos.

Al igual que podemos hacer un análisis clúster para clasificar los objetos en grupos, también podemos hacerlo para clasificar variables. En ese caso se intercambiarían las filas y columnas de la matriz de datos.

3.1.2. Datos

Los datos con los que trabajaremos en cada análisis serán un conjunto de datos de n individuos donde se observarán una serie de variables $x_1, x_2, ..., x_p$. Estos datos estarán estructurados en una matriz X de orden $n \times p$, formada por n casos (filas) y p variables (columnas). Dicha matriz tendrá la forma de la (3.1). A partir de esta matriz se obtendrá otra, llamada matriz de similaridades o de distancias (según el caso), de orden $n \times n$, donde se establece la distancia o similitud entre cada par de individuos.

Antes de realizar el análisis, será importante elegir el tipo de escala.

Las variables cuantitativas requerirán una trasnformación previa, ya que determinadas variables cuantitativas pueden tener un mayor peso que otras por el hecho de que la unidad de medida en la que están, les otorga puntuaciones relativamente altas en comparación con los valores de otras, de manera que podrían llegar a eliminar la influencia que tienen las demás variables.

Una de las transformaciones más populares es la estandarización. Con esto se evitaría el problema de la influencia de las diferentes unidades de medida.

Las variables binarias no suelen ser transformadas. Y las categóricas suelen convertirse en binarias.

3.1.3. Distancia y similitud

El concepto de "similar" vendrá dado por una medida de proximidad o una distancia que cuantificará el grado de semejanza o disimilaridad entre dos elementos.

3.1.3.1. Distancia

Una distancia se considera como una aplicación $d : \mathbb{R}^p \times \mathbb{R}^p \to \mathbb{R}$, dados $x_i = (x_{i1}, \ldots, x_{ip})$ y $x_j = (x_{j1}, \ldots, x_{jp})$ con $i, j = 1, ..., n$ cumpliendo las siguientes propiedades:

1. $d(x_i, x_j) \geq 0$ $\qquad \forall i, j = 1, ..., n$
2. $d(x_i, x_i) = 0$ $\qquad \forall i = 1, ..., n$
3. $d(x_i, x_j) = d(x_j, x_i)$ $\qquad \forall i, j = 1, ..., n$
4. $d(x_i, x_j) \leq d(x_i, x_h) + d(x_h, x_j)$ $\qquad \forall i, j, h = 1, ..., n$

En el caso de que d cumpla todas las propiedades excepto la **4)**, diremos que d es una medida de disimilaridad.

Partiendo de nuestra matriz de datos X, construiremos la matriz de distancias D de orden $n \times n$, donde cada elemento de la matriz d_{ij} se obtiene del valor de $d(x_i, x_j)$ siendo d una distancia, o medida de disimilaridad, y representará el grado de disimilitud o distancia entre el objeto i-ésimo y el j-ésimo.

$$D = \begin{bmatrix} d_{11} & d_{12} & \ldots & d_{1n} \\ d_{21} & d_{22} & \ldots & d_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ d_{n1} & d_{n2} & \ldots & d_{nn} \end{bmatrix} \qquad d_{ij} = d(x_i, x_j) \tag{3.2}$$

La matriz de distancias (3.2) será simétrica debido a que $d_{ij} = d_{ji}$ por la propiedad **3)** de distancia.

Esta matriz, con el mismo conjunto de datos, variará dependiendo de la distancia escogida y de si ha habido alguna transformación o estandarización en las varibales originales.

Cuanta mayor distancia haya entre dos elementos, diremos que son menos parecidos y cuanto menor sea la distancia, más similares.

Hay una infinidad de distancias diferentes conocidas, dependiendo del tipo de variables con las que trabajemos, podremos utilizar unas distancias u otras.

Para el caso en el que todas las variables son cuantitativas, las medidas más utilizadas son las que se basan en normas L_q (o de Minkowski) donde $d(x_i, x_j) = \sqrt[q]{\sum_{k=1}^{p} |x_{ik} - x_{jk}|^q}$, cuanto más grande sea q más influencia tendrá las diferencias entre variables. Una de las más populares es la distancia Euclídea:

Distancia Euclídea (L_2): $d(x_i, x_j) = \sqrt{\sum_{k=1}^{p} (x_{ik} - x_{jk})^2}$.

A pesar de su sencillez, el problema que tiene esta distancia es que es sensible a las unidades de medida de las variables, las diferencias entre variables medidas con valores altos influirán más que las diferencias entre los valores de las variables con valores más bajos (en ocasiones convertirán esta información en redundante). Por lo tanto la distancia Euclídea será recomendable utilizarla cuando las variables estén medidas en unidades similares.

Una de las distancias que no depende de las unidades es la distancia de Canberra:

Distancia de Canberra $d(x_i, x_j) = \sum_{k=1}^{p} \frac{|x_{ik} - x_{jk}|}{|x_{ik} + x_{jk}|}$.

3.1.3.2. Medidas de similitud

Previamente hemos visto que una distancia alta entre dos elementos indica que son diferentes y una baja que son similares, las medidas de similitud funcionan de manera contraria, a medida que aumente el valor, aumentará la similaridad entre los elementos. La mayoría de este tipo de indicadores se basan en coeficientes de correlación o de asociación.

Al comparar dos objetos i, j cuyas variables son todas binarias, podemos construir la siguiente tabla:

$i \setminus j$	1	0
1	a	b
0	c	d

a= Total de 1's coincidentes en el objeto i y j.

b= Total de casos en el que en el objeto i hay un 1 y en el j un 0.

c= Total de casos en el que en el objeto i hay un 0 y en el j un 1.

d= Total de 0's coincidentes en el objeto i y j.

$a + b + c + d = p$ (número total de variables).

Para datos binarios una medida de similitud conocida es el coeficiente de Jaccard:

Coeficiente de Jaccard: $S_{JACCARD}(x_i, x_j) = \dfrac{a}{a+b+c}$.

Podemos observar que el coeficiente de Jaccard no considera las coincidencias de valores 0, por lo que se debería usar para variables binarias asimétricas, donde el valor 1 es más informativo que el valor 0. Por ejemplo en la presencia de una enfermedad poco común como el síndrome de Down, si el valor 1 indica la presencia de esta enfermedad y el 0 la ausencia de esta, dos individuos serán más parecidos si ambos tienen este síndrome, que dos personas que no lo tienen, por lo que la coincidencia del 1 es mucho más significativa que la coincidencia del 0.

Otra medida es la proporción de coincidencias:

Proporción de coincidéncias: $S(x_i, x_j) = \dfrac{a+d}{a+b+c+d}$.

3.1.3.3. Medidas de similitud y distancia entre grupos

El análisis clúster requiere, en general, no solamente el cálculo de la similitud o de las distancias entre objetos iniciales, sino, también la determinación de las distancias o similaridades entre grupos y/o entre grupo y objeto.

Hay varias opciones diferentes para definir la distancia entre grupos, algunas de ellas son:

Distancia mínima

La distancia entre un grupo y un elemento se define como la menor de las distancias entre los objetos del grupo y el elemento exterior que consideramos.

Si llamamos I al grupo formado por los objetos $(i_1, i_2, ...)$ y j al elemento exterior, la distancia entre I y j la definiremos como: $d_G(I, j) = \text{mín}\{d(i, j)/i \in I\}$.

Siguiendo el mismo criterio, definimos la distancia entre dos grupos I y J, como la mínima

distancia entre un elemento de I y otro de J: $d_G(I,J) = \text{mín}\{d(i,j)/i \in I, j \in J\}$

Distancia máxima

También podemos definir la distancia entre un grupo I y un elemento j como el valor máximo de las distancias entre un elemento del grupo I y el elemento j, es decir: $d_G(I,j) = \text{máx}\{d(i,j)/i \in I\}$.

La distancia entre dos grupos I y J será: $d_G(I,J) = \text{máx}\{d(i,j)/i \in I, j \in J\}$

Distancia entre centroides

También podemos definir la distancia entre un grupo I y un elemento j como la distancia entre el centroide de I y el elemento j, es decir, si i es el centroide de I: $d_G(I,j) = d(i,j)$.

De la misma manera la distancia entre un grupo I y un grupo J, será la distancia entre sus centroides, si i es el centroide de I y j el centroide de J: $d_G(I,J) = d(i,j)$.

En cuanto a indicadores de similitud, un ejemplo de medida de similaridad entre los grupos I y J, siendo $I = (i_1, i_2, ..., i_{k_I})$, $J = (j_1, j_2, ..., j_{k_J})$, siendo k_I y k_J el número de elementos de cada grupo respectivamente, sería:

$$S(I,J) = \cos\left[\frac{1}{k_I \cdot k_J}\sum_{i=1}^{k_I}\sum_{j=1}^{k_J}\cos^{-1}(S(i_i, j_j))\right]$$

3.1.4. Métodos de análisis clúster

La creación de los clústers puede ser efectuada de diferentes maneras dependiendo de los algoritmos de cálculo utilizados. Para los mismos datos, en general, métodos distintos darán soluciones distintas.

3.1.4.1. Métodos de partición

En los métodos de partición se parte de un número prefijado de clústers K que no puede ser mayor al número de elementos que tenemos en nuestro conjunto de datos a analizar, es decir $K < n$. Elegido el número de clústers, se etiqueta cada uno con un número $k \in \{1, ...K\}$ y cada objeto es asignado a un clúster, $C(i) = k, i \in \{1, ..., n\}$. Se trata de encontrar una asignación C que minimice la suma de las disimilaridades entre todos los pares de elementos dentro de cada grupo:

$$W(C) = \sum_{k=1}^{K} \sum_{C(i)=k} \sum_{C(j)=k} d(x_i, x_j)$$

Esta optimización solamente sería factible para n y K pequeños, por lo que nos vemos obligados a buscar estrategias iterativas que nos garanticen la convergencia a un óptimo local. En este tipo de estrategias se parte de una asignación inicial y en cada paso se cambia una pequeña parte de las asignaciones de manera que el valor de $W(C)$ disminuya de valor respecto el anterior paso y finalmente en el momento en el que no se produce ninguna disminución del valor de $W(C)$ el algoritmo termina.

Uno de los métos de partición más conocidos y usados es el método de k-medias.

Método de k-medias

El método de k-medias utiliza como medida de disimilaridad la **distancia Euclídea** al cuadrado, por lo que el objetivo es encontrar una asignación C que minimice:

$$W(C) = \sum_{k=1}^{K} \sum_{C(i)=k} \sum_{C(j)=k} d^2(x_i, x_j) = \sum_{k=1}^{K} \sum_{C(i)=k} \sum_{C(j)=k} \sum_{l=1}^{p} (x_{il} - x_{jl})^2$$

La anterior función la podemos minimizar utilizando un algoritmo iterativo descendiente teniendo en cuenta las siguientes propiedades de la media y de la distancia Euclídea. Dado un conjunto de datos $\{x_1, x_2, ..., x_n\}$ se cumple:

1. $\sum_{i=1}^{n} \sum_{j=1}^{n} (x_i - x_j)^2 = \sum_{i=1}^{n} (x_i - \overline{x})^2$

2. $\overline{x} = \arg\min_k \sum_{i=1}^{n} (x_i - k)^2$

Si definimos $\overline{x}_k = (\overline{x}_1^{(k)}, ..., \overline{x}_p^{(k)})$ como el vector de medias asociado con el clúster k-ésimo, aplicando la propiedad **1)**, tendremos:

$$W(C) = \sum_{k=1}^{K} \sum_{C(i)=k} \sum_{C(j)=k} \sum_{l=1}^{p} (x_{il} - x_{jl})^2 = \sum_{k=1}^{K} \sum_{C(i)=k} \sum_{l=1}^{p} (x_{il} - \overline{x}_l^{(k)})^2.$$

Podemos minimizar $W(C)$ resolviendo el problema de optimización ampliado:

$$\min_{C,\{m_{kl}\}} W(C, \{m_{kl}\}) = \sum_{k=1}^{K} \sum_{C(i)=k} \sum_{l=1}^{p} (x_{il} - m_{kl})^2$$

Conociendo por la propiedad **2)** que el mínimo respecto a m_{kl} se obtiene en $\overline{x}_l^{(k)}$.

A partir de aquí hay varios algoritmos para encontrar el mínimo de la función. Uno de los algoritmos más sencillos y que es el originalmente propuesto es el **algoritmo de Lloyds**.

El algoritmo consiste en 3 pasos:

1. Partimos de K objetos, que se toman como los vectores de medias asociados a los K grupos.
2. Cada objeto se le asigna al clúster cuya distancia con la media de dicho clúster sea menor, es decir:

   $C(i) = \arg\min_{1 \leq k \leq K} d(x_i, \overline{x}_k)$
3. Dada la asignación C, recalculamos los nuevos vectores de medias.

Tras el paso 1, los pasos 2 y 3 son iterados hasta que no se produzcan cambios en las asignaciones.

A pesar de que converga, el resultado puede no ser un mínimo absoluto sino un mínimo local. Por ello deberíamos usar el algoritmo varias veces, utilizando diferentes objetos iniciales como vectores de medias cada vez, y elegir la solución con el menor valor de la función objetivo.

3.1.4.2. Métodos jerárquicos

Los métodos jerárquicos parten de una matriz de similitud o de distancias entre los elementos del conjunto de datos y construyen una jerarquía basándose en las distancias o similitudes, es decir, los grupos están anidados en los de pasos anteriores.

Fundamentalmente los métodos jerárquicos se subdividen en dos tipos:

- **Métodos aglomerativos**: Van fusionando sucesivamente grupos en cada paso.
- **Métodos divisivos**: Van desglosando sucesivamente en grupos el conjunto total de datos.

Los aglomerativos son más sencillos y son los más utilizados, por esa razón dedicaremos más atención a este tipo de métodos jerárquicos.

3.1.4.2.1. Métodos jerárquicos aglomerativos .

En este tipo de métodos, también conocidos como ascendentes, se comienza el análisis con tantos clústers como elementos tenga la muestra y a partir de estos clústers iniciales, que solamente están formados por un elemento cada uno, se van formando nuevos grupos de forma ascendente hasta que al final del proceso todos están englobados en un mismo clúster.

A diferencia de los métodos de partición, en este tipo de métodos no se efectuarán reasignaciones, es decir, no se podrán realizar cambios en los resultados en pasos sucesivos. Una vez dos elementos o clústers se hayan unido en un nuevo clúster, permanecerán de esta manera hasta el final del proceso.

Todos los algoritmos de aglomeración tienen la misma estructura, solamente se diferencian en la forma de calcular la distancia o similitud entre clústers.

Partiendo de una matriz de similitud o de distancias entre objetos $D = [d_{ij}]_{p\times p}$ con $d_{ij} = d(x_i, x_j)$, denotamos por D_G a la matriz de distancias entre grupos, que inicialmente será igual a la matriz de distancias entre objetos, ya que al principio tenemos tantos clústers como objetos, pero posteriormente a cada paso se irá reduciendo. Se siguen 3 pasos:

1. Se empieza con tantos clústers como objetos tenga la muestra, es decir, n grupos unitarios en los que C_i correspondería al clúster formado por el objeto i-ésimo, y de una matriz de distancias entre grupos que inicialmente está formada por las distancias entre los clústers iniciales formados únicamente por un objeto cada uno, es decir, $D_G = D$.

2. Se seleccionan los dos elementos más cercanos en la matriz de distancias:

$$(i^*, j^*) = \arg\min_{i,j} D_G(i,j) \text{ con } i \neq j$$

En el caso de trabajar con medidas de similitud, considerando S_G la matriz de similitud entre grupos, los más cercanos, es decir, los más similares, serían:

$$(i^*, j^*) = \arg\max_{i,j} S_G(i,j) \text{ con } i \neq j$$

3. En la matriz D_G se sustituyen las dos filas (columnas) que corresponden a los elementos y/o conjuntos i^* y j^*, obtenidos en el paso anterior, por una nueva fila (columna) cuyo índice es el $k = (i^*, j^*)$:

$$D_G(k = (i^*, j^*), l) = d_G(C_{i^*} \cup C_{j^*}, C_l)$$

con l desde el índice de la primera columna hasta la última.

Notar que al realizar este paso, la matriz sigue siendo cuadrada pero con una dimensión de 1 unidad menor.

Los pasos 2 y 3 se repiten hasta que todos los elementos queden agrupados en el mismo clúster, o lo que es lo mismo, que la matriz D_G sea regular de dimensión 1.

En el paso 3, dependiendo de cómo calcular la distancia entre clústers, tenemos varios algoritmos. Algunos de ellos son:

- **Encadenamiento simple o vecino más próximo:** Este método utiliza la **distancia mínima** entre grupos, es decir, la distancia entre los clústers viene dada por la mínima distancia entre las componentes de un clúster y del otro.

$$d_G(C_i, C_j) = \text{mín}\{d(x_i, x_j)/x_i \in C_i, x_j \in C_j\}$$

En el paso 3) tendremos $d_G(C_i \cup C_j, C_l) = \text{mín}(D_G(i,l), D_G(j,l))$.

En el caso de trabajar con medidas de similitud en vez de con distancias, en las fórmulas en vez de mín será máx.

- **Encadenamiento completo o vecino más alejado:** Este método utiliza la **distancia máxima** entre grupos, es decir, la distancia entre los clústers viene dada por la máxima distancia entre las componentes de un clúster y del otro.

$$d_G(C_i, C_j) = \text{máx}\{d(x_i, x_j)/x_i \in C_i, x_j \in C_j\}$$

En el paso 3) tendremos $d_G(C_i \cup C_j, C_l) = \text{máx}(D_G(i,l), D_G(j,l))$

- **Encadenamiento medio entre grupos:** Este método mide la distancia entre dos clústers calculando la media ponderada entre los objetos de cada grupo.

  En el paso 3 tendremos $d_G(C_i \cup C_j, C_l) = \frac{n_i}{n_i+n_j} D_G(i,l) + \frac{n_j}{n_i+n_j} D_G(j,l)$ con n_i, n_j el tamaño de los clústers.

Cabe destacar que dependiendo del método utilizado, obtendremos, por lo general, soluciones distintas, sobre el mismo conjunto de datos.

3.1.4.2.2. El dendrograma .

El dendrograma es una representación gráfica del resultado del proceso en forma de árbol binario. Cada nodo representa, mediante sus dos hijos, la unión de dos clústers y la altura de cada nodo es proporcional al valor de la disimilaridad entre sus dos hijos. El nodo inicial representa todo el conjunto de datos y los nodos terminales representan los n objetos iniciales y están dibujados a altura cero. En la figura 3.1 podemos ver un ejemplo de un dendrograma.

En el caso de cortar el dendograma a un nivel de distancia dado, obtendremos una clasificación del número de grupos existente a ese nivel y los objetos que los forman.

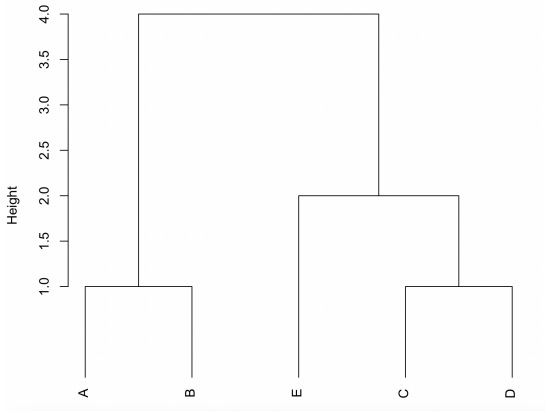

Figura 3.1: Dendrograma

3.2. Análisis estadístico implicativo (ASI)

3.2.1. Introducción

El análisis estadístico implicativo, ASI por sus siglas en francés, es un método de análisis no simétrico de datos que nos permite, a partir de un conjunto de datos, la extracción y estructuración del conocimiento en forma de reglas y normas. Su origen viene de la *modelización estadística de la cuasi-implicación*: si una variable o conjunto de variables **a** es observada en la población, entonces generalmente también lo es la variable **b** [4, 8].

3.2.2. Sistema informático CHIC

El sístema informático CHIC es un software de análisis de datos, escrito en C++, desarrollado en un inicio para la aplicación del ASI.

Este software realiza cálculos estadísticos sobre las variables, obteniendo como resultado distintos gráficos, según los índices de similitud y el tipo de clasificación, jerárquica, implicativa, cuasi-implicativa o inclusiva.

El software muestra dos tipos de árboles y un grafo:

- **Árbol de similaridad:** Es el más conocido y en el que centraremos el estudio en estre proyecto, se forma a partir del índice de similaridad definido por Lerman [6], el cual permite construir una jerarquia ascendente.
- **Árbol jerárquico orientado:** Se forma mediante la intensidad de la implicación.
- **Grafo implicativo:** Permite al usuario seleccionar las reglas de asociaciones y las variables deseadas.

Además de mostrar los diferentes modos de representación, CHIC muestra los cálculos estadísticos realizados.

Mediante estos gráficos, podemos formar clases de cuasi-equivalencia entre las variables tratadas y llegar a conclusiones sobre la muestra estudiada.

3.2.3. Análisis clasificatorio

El análisis clasificatorio es una técnica de análisis exploratorio cuyo objetivo es el mismo que el del análisis clúster estándar de la sección 3.1, que consiste en revelar las agrupaciones naturales dentro de un conjunto de datos.

Para poder realizar estas agrupaciones, se necesita una medida de similaridad para evaluar la similitud entre objetos.

Para la realización de este análisis mediante el sistema informático CHIC, se parte considerando un conjunto I formado por n objetos y un conjunto A formado por p características, $A = \{a_1, a_2, ..., a_p\}$, y suponiendo que:

$$A_i = \{x \in I / a_i(x) = 1\}, \;\; Card(I) = n \text{ y } Card(A_i) = n_{a_i}$$

El ASI, a diferencia de la mayoría de métodos de clasificación, calcula los índices de similaridad en términos de una probabilidad, cuyo cálculo depende del modelo asumido para la variable aleatoria $Card(X_i \cap X_j)$. Las leyes de distribución que puede seguir esta variable, $Card(X_i \cap X_j)$ son: Hipergeométrica, Binomial y Poisson. El programa CHIC solamente deja seleccionar entre Binomial y Poisson.

3.2.3.1. Árbol de Similaridad

Para formar el árbol de similaridad, se calculan los índices de similaridad definidos por Lerman [6] entre cada par de variables (a_i, a_j):

$$s(a_i, a_j) = Pr[Card(X_i \cap X_j) \leq K]$$

donde $K = Card(A_i \cap A_j)$ es el número de copresencias (coincidencias de 1's) que hay entre las variables a_i y a_j.

El cálculo de Pr dependerá de la ley de probabilidad asumida, Binomial o Poisson. Independientemente de la ley asumida, Poisson o Binomial, CHIC realiza una aproximación de éstas a la Normal devolviendo:

$$s(a_i, a_j) = Pr\left[\frac{Card(X_i \cap X_j) - \frac{n_{a_i} \times n_{a_j}}{n}}{\sqrt{\frac{n_{a_i} \times n_{a_j}}{n}}} \leq K_c\right] = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{K_c} e^{\frac{1}{2}x^2} \, dx,$$

siendo $K_c := \dfrac{K - \dfrac{n_{a_i} \times n_{a_j}}{n}}{\sqrt{\dfrac{n_{a_i} \times n_{a_j}}{n}}}$.

Los índices $s(a_i, a_j)$ representan las similaridades entre cada par de variables (a_i, a_j) a nivel cero de la jerarquía.

El índice de similaridad entre dos clústers de variables C_1 y C_2, cada uno con r_1 y r_2 variables, respectivamente, viene definido, por Lerman [6], como:

$$s(C_1, C_2) = (\text{máx}\{s(a_i, a_j)/a_i \in C_1, a_j \in C_2\})^{r_1 \times r_2} \tag{3.3}$$

Por lo que las similaridad entre una clase con dos variables (a_i, a_j) y otra variable a_k será:

$$s((a_i, a_j), a_k) = (\text{máx}\{s(a_i, a_k), s(a_j, a_k)\})^{2 \times 1}$$

El proceso a seguir es muy parecido al que aplicabamos en el análisis clúster clásico, apartado 3.1.4.2.1. Construímos una primera matriz con los índices de similaridad $s(a_i, a_j)$, obtenidos a partir de todas las combinaciones posibles entre las variables, a esta matriz S la podemos llamar matriz de similitudes, luego definimos una matriz S_G que representará a la matriz de similitud entre grupos y seguimos los siguientes pasos:

1. Consideramos tantos clústers como número de variables totales que tenemos, cada uno de estos clústers inicialmente contendrá una única variable, por lo que la matriz de similitud entre grupos y la matriz de similitud coincidirán en este paso, es decir $S_G = S$.
2. Se busca dentro de la matriz S_G el par de grupos que tengan el mayor índice de similaridad:

   $$(i^*, j^*) = \arg \max_{i,j} S_G(i, j) \text{ con } i \neq j$$

3. En la matriz S_G sustituimos las dos filas (columnas) etiquetadas por los clústers obtenidos del paso 2, i^* y j^*, por una fila (columna) etiquetada por un clúster que une a los dos anteriores y mediante (3.3) calculamos los nuevos índices de similaridad entre este clúster y los demás:

   $$S_G(k = (i^*, j^*), l) = s(k, l)$$

   con l recorriendo cada uno de los clústers que hay en la matriz.

Los pasos **2** y **3** se van iterando hasta que solo queda un clúster en la matriz S_G, este clúster engloba a todas las variables. A cada iteración se sube un nivel en la jerarquía.

Ejemplo

Nuestro conjunto de datos está formado por una serie de 41 animales y 75 tipos de características en francés, en un archivo llamado *animaux.csv* que viene proporcionado tanto en el software CHIC como en la librería *rchic*, y lo visualizamos en la sección del apéndice A.2. En este ejemplo, para simplificar, utilizaremos un subconjunto de datos formado por las siguientes 6 variables binarias: Affectueux (Afectuoso), Agile (Ágil) , Agressif (Agresivo), Angoissant (Aterrador), Attirant (Atractivo) y Beau (Hermoso), y 41 observaciones, $n = 41$. Para estas variables, $n_{AFF} = 5$, $n_{AGI} = 12$, $n_{AGR} = 19$, $n_{ANG} = 17$, $n_{ATT} = 9$, $n_{BEA} = 20$ representan la cantidad de animales que son cualificados por estos adjetivos, respectivamente.

En la tabla 3.1 podemos ver el valor de las variables para cada observación, y en la tabla 3.2 las copresencias $Card(A_i \cap A_j)$ y los índices de similaridad $s(a_i, a_j)$ a nivel cero de la jerarquía, calculado bajo la aproximación de la Normal.

	AFF	AGI	AGR	ANG	ATT	BEA
X1	0	1	0	1	1	1
X2	0	0	0	0	0	1
X3	0	0	0	1	0	0
X4	0	1	1	0	1	1
X5	0	0	1	0	1	0
X6	0	0	1	0	0	1
X7	0	1	0	0	0	1
X8	1	1	0	0	1	1
X9	1	0	0	0	0	0
X10	0	0	0	0	0	1
X11	0	0	1	1	0	1
X12	0	0	0	1	0	0
X13	0	0	1	1	0	0
X14	0	1	1	1	0	0
X15	0	0	0	0	1	1
X16	0	0	0	0	1	0
X17	0	1	0	0	0	1
X18	0	1	1	1	0	0
X19	0	0	0	0	0	0
X20	0	0	1	0	1	1
X21	0	0	1	1	0	1
X22	0	1	1	0	0	1
X23	0	0	0	0	0	0
X24	1	0	1	0	0	0

	AFF	AGI	AGR	ANG	ATT	BEA
X25	0	0	1	1	0	0
X26	1	0	0	1	1	1
X27	0	1	1	1	0	0
X28	0	0	0	0	0	0
X29	0	0	1	0	0	0
X30	0	0	0	0	0	0
X31	0	0	1	1	0	0
X32	0	0	0	0	0	1
X33	0	0	1	1	0	1
X34	1	1	0	0	0	0
X35	0	0	1	0	0	0
X36	0	1	1	1	1	1
X37	0	0	0	0	0	0
X38	0	0	0	0	0	0
X39	0	1	1	1	0	1
X40	0	0	0	1	0	1
X41	0	0	0	0	0	1

Tabla 3.1: Matriz de datos

Variables	$Card(A_i \cap A_j)$	$s(a_i, a_j)$
(AFF,AGI)	2	0.6713206
(AFF,AGR)	1	0.1934516
(AFF,ANG)	1	0.2280347
(AFF,ATT)	2	0.8054904
(AFF,BEA)	2	0.3893120
(AGI,AGR)	7	0.7291449
(AGI,ANG)	6	0.6769701
(AGI,ATT)	4	0.7999824
(AGI,BEA)	7	0.6821808
(AGR,ANG)	12	0.9290247
(AGR,ATT)	3	0.2832344
(AGR,BEA)	8	0.3384854
(ANG,ATT)	4	0.5552295
(ANG,BEA)	9	0.5970124
(ATT,BEA)	7	0.8935322

Tabla 3.2: Valor de las copresencias e índices de similaridad de cada par de variables

A partir de la tabla 3.2 construimos la matriz de similitud a nivel cero de la jerarquía, representada en la tabla 3.3.

	AFF	**AGI**	**AGR**	**ANG**	**ATT**	**BEA**
AFF	1	0.6713206	0.1934516	0.2280347	0.8054904	0.3893120
AGI	0.6713206	1	0.7291449	0.6769701	0.7999824	0.6821808
AGR	0.1934516	0.7291449	1	**0.9290247**	0.2832344	0.3384854
ANG	0.2280347	0.6769701	**0.9290247**	1	0.5552295	0.5970124
ATT	0.8054904	0.7999824	0.2832344	0.5552295	1	0.8935322
BEA	0.3893120	0.6821808	0.3384854	0.5970124	0.8935322	1

Tabla 3.3: Matriz de similitud a nivel 0 de la jerarquía

El mayor valor es el 0,9290247, que es el valor del índice de similitud entre la variable **AGR** y **ANG**, por lo tanto en el siguiente nivel sustituímos las dos filas (columnas) etiquetadas por **AGR** y **ANG** por una fila (columna) etiquetada por una clase formada por las variables **AGR** y **ANG** y calculamos los índices de similitud entre esta clase y las demás variables, mediante la fórmula 3.3 y entonces obtenemos la tabla 3.4.

	AFF	**AGI**	**{AGR,ANG}**	**ATT**	**BEA**
AFF	1	0.6713206	0.05199981	0.8054904	0.3893120
AGI	0.6713206	1	0.5316523*	0.7999824	0.6821808
{AGR,ANG}	0.05199981	0.5316523*	1	0,3082798	0.3564238
ATT	0.8054904	0.7999824	0.3082798	1	**0.8935322**
BEA	0.3893120	0.6821808	0.3564238	**0.8935322**	1

Tabla 3.4: Matriz de similitud a nivel 1 de la jerarquía

El valor 0.5316523 de la tabla 3.4 se ha obtenido de aplicar 3.3:

$$\begin{aligned} s(\{AGR, ANG\}, AGI) &= (max\{s(AGR, AGI), s(ANG, AGI)\})^2 \\ &= (max\{0{,}7291449, 0{,}6769701\})^2 = 0{,}7291449^2 = 0{,}5316523 \end{aligned}$$

Ahora el valor máximo es 0.8935322, por lo que se unirán las variables **ATT** y **BEA**, aplicando el mismo procedimiento obtendremos la matriz de similitud a nivel 2, 3.5

El valor 0.1270379 de la tabla 3.5 viene de aplicar 3.3:

	AFF	**AGI**	**{AGR,ANG}**	**{ATT,BEA}**
AFF	1	**0.6713206**	0.05199981	0.6488148
AGI	**0.6713206**	1	0.5316523	0.6399719
{AGR,ANG}	0.05199981	0.5316523	1	0.1270379**
{ATT,BEA}	0.6488148	0.6399719	0.1270379**	1

Tabla 3.5: Matriz de similitud a nivel 2 de la jerarquía

$$
\begin{aligned}
&s(\{AGR, ANG\}, \{ATT, BEA\}) \\
&= (max\{s(AGR, ATT), s(AGR, BEA), s(ANG, ATT), s(ANG, BEA)\}^4 \\
&= (max\{0{,}2832344, 0{,}3384854, 0{,}5552295, 0{,}5970124\})^4 = 0{,}5970124^4 = 0{,}1270379
\end{aligned}
$$

En el nivel 3 tendremos la matriz 3.6

	{AFF,AGI}	**{AGR,ANG}**	**{ATT,BEA}**
{AFF,AGI}	1	0.2826542	**0.4209607**
{AGR,ANG}	0.2826542	1	0.1270379
{ATT,BEA}	**0.4209607**	0.1270379	1

Tabla 3.6: Matriz de similitud a nivel 3 de la jerarquía

En el nivel 4 tendremos la matriz 3.7

	{{AFF,AGI},{ATT,BEA}}	**{AGR,ANG}**
{AFF,AGI,ATT,BEA}	1	0.0798934
{AGR,ANG}	0.0798934	1

Tabla 3.7: Matriz de similitud a nivel 4 de la jerarquía

Y finalmente en el nivel 5 tendríamos una única clase formada por todas las variables iniciales.

Mediante el uso de *rchic* en R podemos visualizar el árbol de la figura 3.2.

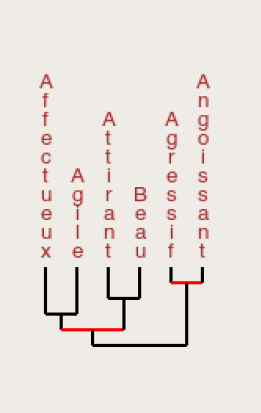

Figura 3.2: Árbol de similitud

3.3. Conexión entre ASI y R

3.3.1. *rchic*

El código del programa CHIC se ha portado parcialmente a R en un paquete llamado *rchic*, este paquete sigue en desarrollo por parte de Raphaël Couturier [2].

El paquete *rchic*, dado un conjunto de datos, permite visualizar el dendrograma del árbol de similitud que se forma a partir de estos datos. Además, guarda la matriz de similaridad calculada en una variable R de la clase matriz y un conjunto con el nombre de las variables.

3.3.2. Árbol de similitud y la clase *hclust*

Para poder usar funciones interesantes de R con el árbol de similitud de ASI, es necesario convertirlo en un objeto de clase *hclust* [10]. Esto no lo hace *rchic*, hasta el momento. Para conseguir convertir el árbol en un objeto de la clase *hclust* es necesario la descripción del árbol: por una parte, la secuencia de variables y/o clústers que se une en cada paso del árbol y por otra parte, el valor del índice de similaridad de las variables y/o clústers que se han unido en cada paso.

La clase *hclust* tiene los siguientes componentes:

- **merge**: Una matriz $(n-1) \times 2$, siendo n el número total de objetos o variabales que

queremos agrupar, en la que la fila i-ésima indica lo ocurrido en el paso i-ésimo, si un número es negativo quiere decir que en la unión participa una variable (o un objeto, dependiendo del tipo de análisis que estemos haciendo) y si el número es positivo, se está uniendo un grupo previamente obtenido.

- **height**: Un conjunto de $n-1$ valores crecientes que refieren al valor de la distancia entre los dos grupos unidos en cada paso.
- **order**: un vector dando la permutación de las variables (o elementos) originales para dibujar el dendograma de forma que no hayan cruces.
- **labels**: Un vector de caracteres de etiquetas de las variables (o elementos)
- **method**: El método usado
- **dist.method**: La distancia usada entre variables (o elementos)

Por lo que hemos de obtener las anteriores componentes a partir de los datos obtenidos en *rchic* para convertir el árbol de similitud en un objeto *hclust*.

En el capítulo siguiente se describe detalladamente cómo se implementa dicha transformación, cuyo resultado final se incluye en el Anexo A.1.

Capítulo 4

Resultados

4.1. Script para convertir un árbol de similitud en un objeto *hclust*

Para crear la función que convierte el árbol de similitud ASI en un objeto *hclust* en R, nos ha servido de guía el árticulo [7] en el que explican los pasos que han realizado para transformar "manualmente" el árbol en objeto *hclust*.

Los pasos seguidos a la hora de escribir el código han sido:

1. Crear un vector de tamaño p, el número de variables, con los números de -1 hasta $-p$. Estos números identifican a cada variable inicial, es decir el $-i$ hace referencia a la variable i-ésima. En el momento en el que una variable forme parte de una unión, la eliminaremos sustituyendo su número dentro del vector por cero. Nosotros la nombramos como *individuals*

2. Crear la matriz *merge* de tamaño $(p-1) \times 2$ en la que la primera fila hará referencia al primer clúster (de mínimo dos variables) creado, la fila i-ésima hará referencia al i-ésimo clúster creado. En cada fila irán dos números que explicarán la unión que se ha realizado, si un número es negativo, querrá decir que se ha unido una variable, si el número es positivo querra decir que se ha unido un clúster anteriormente formado, por lo que si se unen dos variables, la fila contendrá dos números negativos, si se unen dos clústers, la fila contendrá dos números positivos y si se unen un clúster y una variable, la fila contendrá un número negativo y uno positivo.

3. Crear el vector *heights* donde se esperan valores de distancia, por lo que los valores de

los índices de similitud entre las uniones realizadas en cada nivel los transformaremos en distancias mediante la fórmula: $d = \sqrt{1-s}$, siendo s el valor del índice de similitud.

4. Crear una lista en la que cada vez que ocurra una unión añadiremos el número total de uniones realizadas, es decir, el nivel en el que nos encontramos. Esta lista nos servirá para saber que clústers se pueden unir, o lo que es lo mismo, los que aún no se han unido. En el momento en el que un clúster forme parte de una unión, pondremos el número que lo identifica, es decir, el nivel en el que se ha creado, a 0. Nosotros la nombramos como *clusters_indices*

5. Crear una lista en la que en cada posición irá una lista de números negativos que harán referencia a las variables contenidas, es decir, en la posición i-ésima estarán los números negativos correspondientes a las variables que forman parte del clúster i-ésimo. Al igual que en la lista anterior, en el momento en el que un clúster forme parte de una unión, su correspondiente lista de variables se vaciará, ya que no se volverá a utilizar, pero antes de vaciarla copiaremos los valores de la lista en la lista del nuevo clúster formado. Nosotros la nombramos como *clusters*

6. Para obtener los números que irán en la matriz *merge*, recorremos la lista *individuals* y la lista *clusters* y calculamos los índices de similitud entre todos los pares de clústers y variables que estén en las listas mencionadas a partir de la matriz de similitud, que nos viene dada como parámetro, y la fórmula 3.3 y nos quedamos con el valor máximo, entonces lo pasamos a distancia y lo añadimos al vector *heights* y en la matriz *merge* añadimos los dos números correspondientes a los participantes de la unión, si se ha unido una variable, añadimos el número negativo que le identifica y en el vector *individuals* lo ponemos a cero, si se ha unido un clúster, añadimos el número que corresponde al nivel en el que se creó el clúster y en el vector *clusters_indices* ponemos ese valor a cero y quitamos su lista de variables dentro de la lista *clusters*, pero previamente copiamos esta lista de variables a la lista del nuevo clúster formado.

7. El paso anterior se repetiría hasta completar la matriz *merge* y entonces nos quedaría por obtener las componentes *labels* y *orders*.

8. La componente *labels* la obtendriamos de la cabecera de la matriz de similitud que nos viene como parámetro.

9. El vector *orders* lo obtendríamos inicializando el vector con los valores crecientes del 1 hasta el n. Con esto ya tendríamos un objeto *hclust* con la componente *orders* erronea. Para tener la componente correcta, este objeto lo transformaríamos a un objeto de la clase dendrograma mediante la funcion *as.dendrogram*, en el que la componente *orders* es la correcta, por lo que cambiaríamos nuestra componente *orders* de nuestro objeto *hclust* por la obtenida en la clase dendrograma.

4.2. Aplicación del script para la comparación de árboles ASI con dendrogramas del Análisis clúster

En esta sección vamos a comparar fácilmente los dendrogramas resultantes del ASI y del clúster tradicional, gracias al uso de nuestro script.

El conjunto de datos que vamos a utilizar es el procedente del archivo Animaux (completo), disponible en la sección del apéndice A.2.

Vamos a comparar el dendrograma obtenido mediante el método de similaridad ASI con el dendrograma obtenido por el método jerárquico aglomerativo que utiliza el coeficiente de Jaccard con el algoritmo de encadenamiento simple y el dendrograma obtenido utilizando la proporción de coincidéncias con encadenamiento simple.

El dendrograma obtenido mediante el método de similaridad ASI, gracias al uso de nuestra función, es el de la figura 4.1

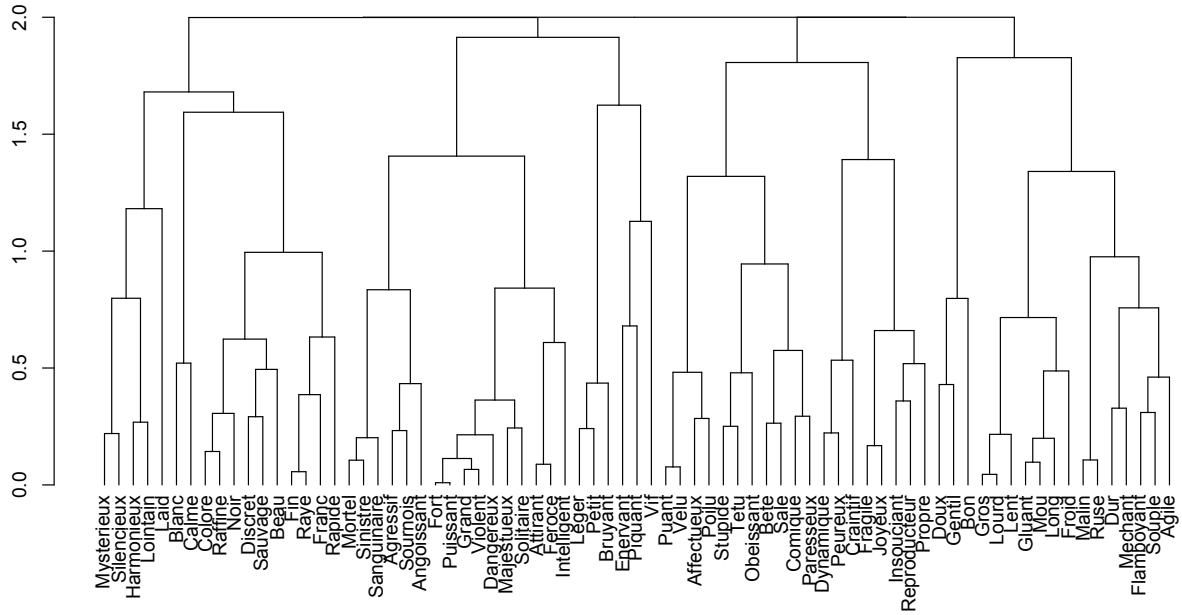

Figura 4.1: Dendrograma del árbol de similitud ASI

El dendrograma obtenido mediante la función *hclust* de R y la matriz de similitud construída a partida de la medida de similitud de Jaccard es el de la figura 4.2

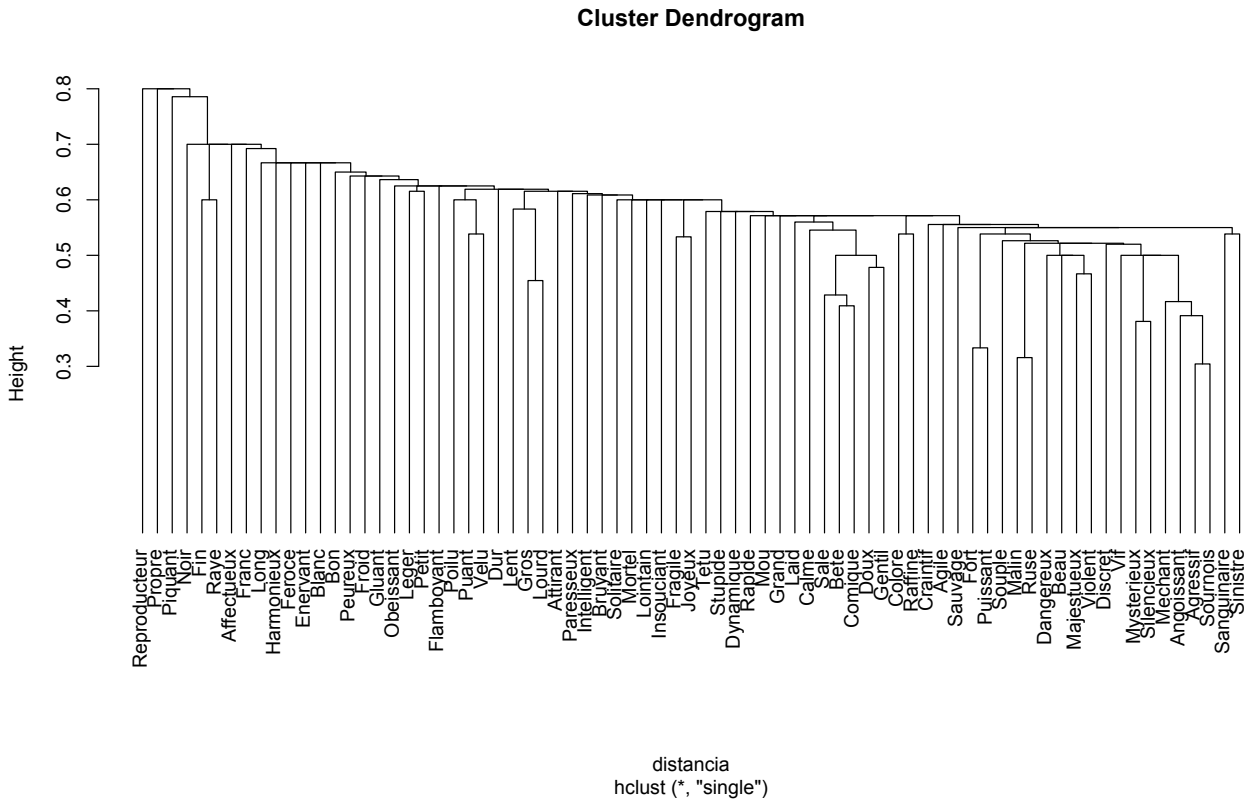

Figura 4.2: Dendrograma utilizando la medida de Jaccard

El dendrograma obtenido mediante la proporción de coincidencias se muestra en la figura 4.3, donde podemos apreciar varios clústers unidos en un mismo clúster, esto es debido a coincidencias entre distancias.

Y mediante el paquete *dendextend* de R [3], su función *tanglegram* nos permite tener una comparación visual de dos dendrogramas. Pone los dendrogramas uno en frente del otro, con las mismas etiquetas, y une mediante líneas las etiquetas idénticas, marcando en color los pares de variables que se han unido en ambos lados y marca en lineas discontinuas las ramas distintas.

La comparación entre el dendrograma de la figura 4.1 y el de la figura 4.2 se muestra en la figura 4.4 y la comparación con la figura 4.3 se muestra en la figura 4.5, donde en ambas comparaciones podemos observar que cada método hace una agrupación muy distinta a la otra.

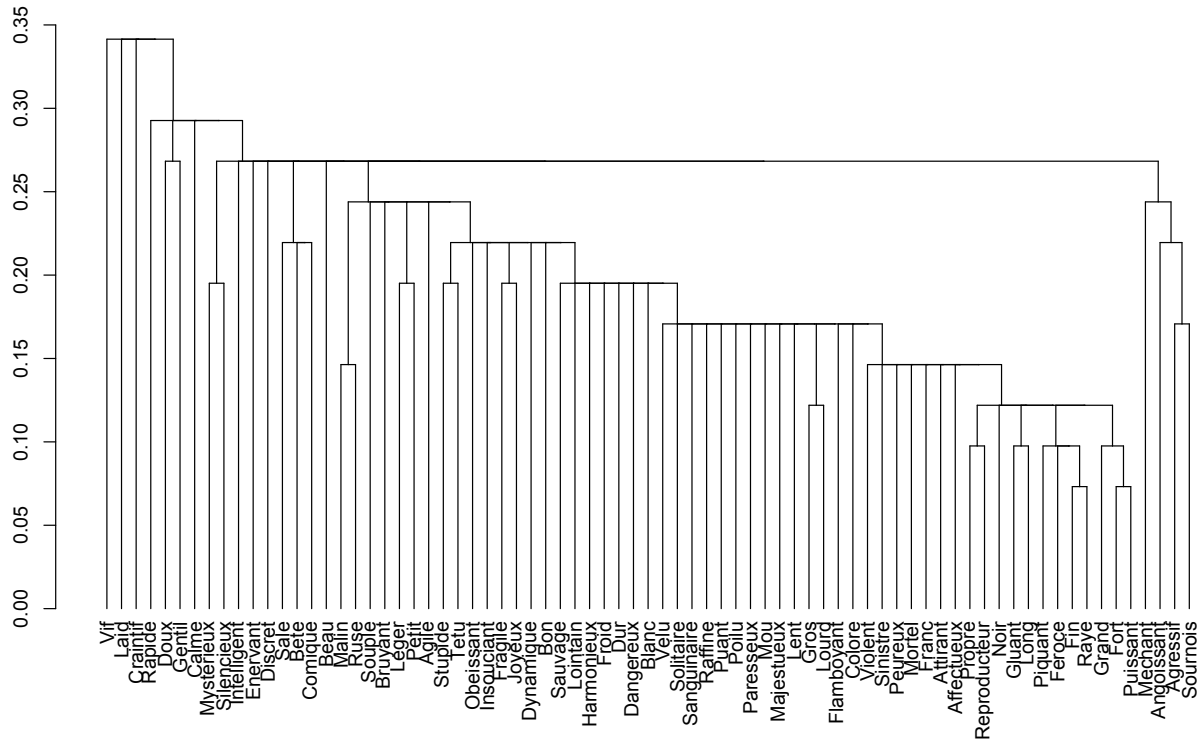

Figura 4.3: Dendrograma utilizando la medida de proporción de coincidencias

En el artículo [7] los autores comparan arboles del ASI con arboles de otra técnica clúster de variables llamada ClustOfVars [1].

Estas comparaciones permiten que la comunidad científica conozca una alternativa más de los análisis clúster, como es el enfoque ASI, y pueda generar interés por su uso en aplicaciones donde no se habría utilizado.

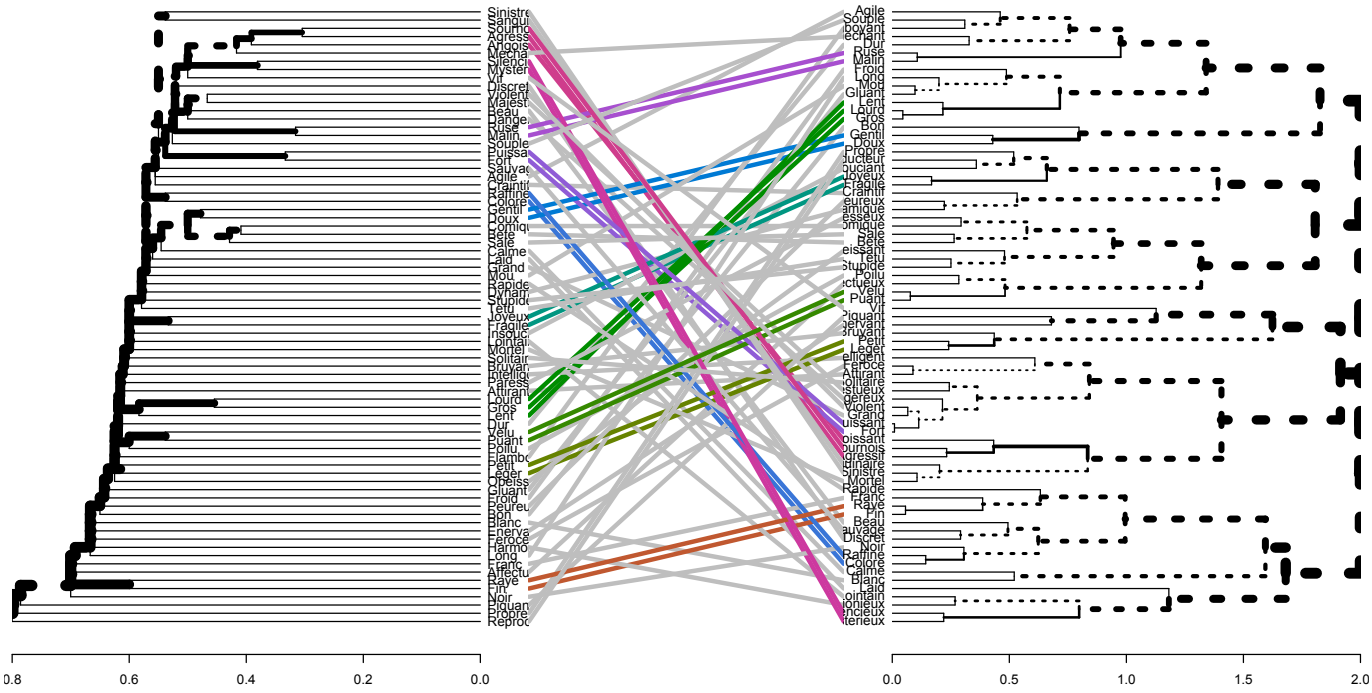

Figura 4.4: Comparación entre el dendrograma de Jaccard y el ASI mediante la función *tanglegram*

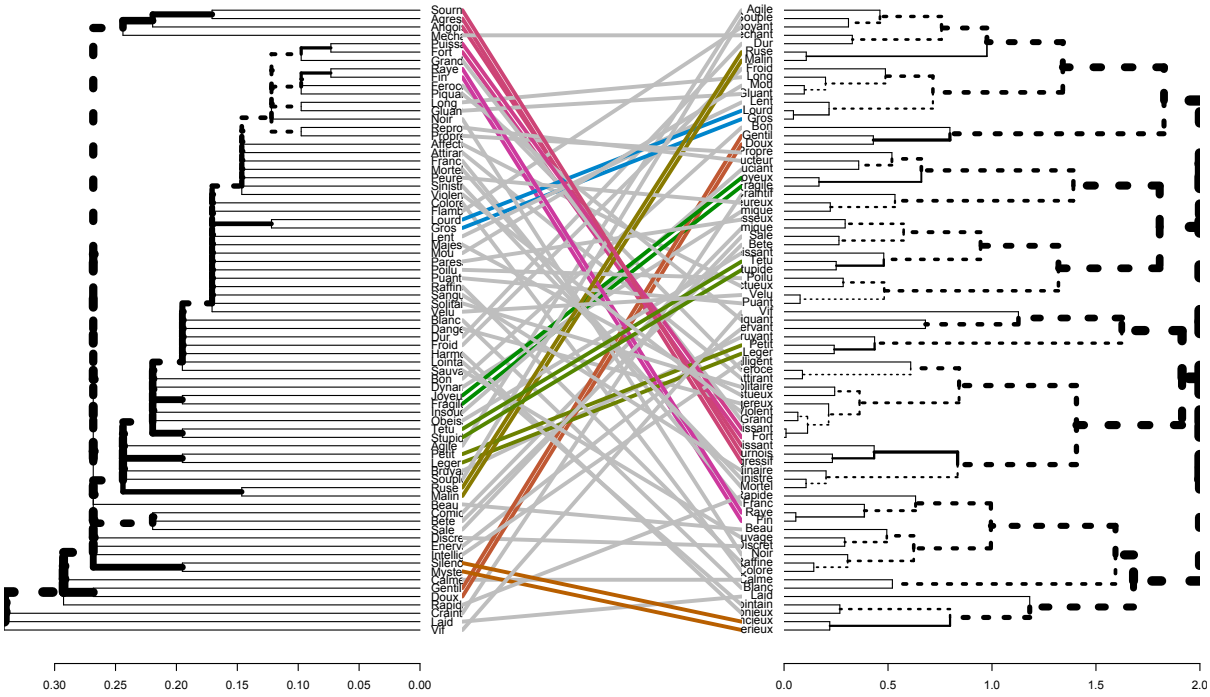

Figura 4.5: Comparación entre el dendrograma de proporción de coincidencias y el ASI mediante la función *tanglegram*

Capítulo 5

Conclusiones

En el presente TFG hemos descrito brevemente el análisis clúster jerárquico, las principales variantes, en cuanto a elección de distancias y métodos de aglomeración, la propuesta del Análisis Estadístico Implicativo y el software que permite su aplicación (CHIC y la librería *rchic* de R). Como aportación significativa, en este TFG hemos programado un pequeño script que aprovecha los cálculos de la función *rchic* (que solo dibuja el árbol de similaridad resultante de un análisis ASI), para definir el resultado como objeto de la clase *hclust* de R. Gracias a ello, el árbol puede procesarse más allá, aprovechando utilidades de librerías de R como *dendextend* (que comparan dendrogramas, ver [3]).

Con este trabajo esperamos haber ayudado a la difusión de la metodología ASI para análisis clúster, por poner sus árboles de similaridad más al alcance de la gran comunidad de usuarios de R.

Bibliografía

[1] Chavent, M., Kuentz, V., Liquet, B. y Saracco, J., 2017. Package 'ClustOfVar'. [online] Cran.r-project.org. Disponible en `https://cran.r-project.org/web/packages/ClustOfVar/ClustOfVar.pdf` [Acceso 5 Junio 2022].

[2] Couturier, R., (2015). GitHub - rcouturier/Rchic: CHIC for R. [online] GitHub. Disponible en `https://github.com/rcouturier/Rchic` [Acceso 5 Junio 2022].

[3] Galili, T., (2021). Introduction to dendextend. [online] Cran.r-project.org. Disponible en `https://cran.r-project.org/web/packages/dendextend/vignettes/dendextend.html` [Acceso 5 Junio 2022].

[4] Gras, R., Almouloud, S-AG., Bailleul, M., Larher, A., Polo, M., Ratsimba-Rajohn, H., Totohasina, A., (1996). L'implication statistique, nouvelle méthode exploratoire de données. Applications à la didactique. Grenoble :La Pensée Sauvage éditions.

[5] Härdle, W. y Simar, L., (2007). Applied Multivariate Statistical Analysis. Springer.

[6] Lerman, I.C., (1970). Sur l'analyse des donnéees préealable à une classification automatique (proposition d'une nouvelle mesure de similaritée). Mathématiques, Informatique et Sciences Humaines, (32):5–15.

[7] Nieto, G. y Gregori, P. (2021) Comparison of two cluster analysis of variables: statistical implicative analysis vs ClustofVar, en *Analyse statistique implicative, Analyses quali-quantitatives des liens orientés entre variables et/ou groupes de variables*, Editores: J.-C. Régnier, R. Gras, A. Bodin, R. Couturier y G. Vergnaud. ISBN 978-2-9562045-5-8. Disponible en `https://sites.univ-lyon2.fr/asi/11/pub/ASI11_ISBN_978-2-9562045-5-8_NUMERIQUE2021.pdf`

[8] Orús, P., Zamora, L. y Gregori, P., (2010). Teoría y aplicaciones del análisis estadístico implicativo. [Castellón de la Plana]: Departamento de Matemáticas, Universitat Jaume I de Castellón, pp.77-85.

[9] Peña, D., (2002). Análisis de Datos Multivariantes. McGraw-Hill.

[10] R Core Team (2021). R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria. Disponible en `https://www.R-project.org`.

Anexo A

Anexo I

A.1. Código R para convertir un árbol de similitud en un objeto *hclust*

La función *hclust2022* devuelve un objeto *hclust* y tiene 1 solo parámetro, este parámetro es la matriz de similitud que calcula la función *rchic* de la librería del mismo nombre.

```
hclust2022 <- function(a) {
    b <- names(a[1,])
    individuals<-c(-1:-length(b))
    clusters<-list()
    clusters_indices<-c()
    A<-matrix(ncol=2,byrow=TRUE)
    contador=1

    while(contador<length(b)){

        max=0
        pair=c(0,0)

        for (clust1 in clusters_indices[clusters_indices>0]){
            r1=length(clusters[[clust1]])
            for ( i in clusters[[clust1]]){
                i=-i
```

```
        for(j in individuals[individuals<0]){
            j=-j
            valor=a[i,j]^r1
            if(valor>=max){
                max=valor
                pair=c(clust1,-j)
                }
            }

            for (clust2 in clusters_indices[clusters_indices>clust1]){
                r2=length(clusters[[clust2]])
                for(j in clusters[[clust2]]){
                    j=-j
                    valor=a[i,j]^(r1*r2)
                    if(valor>=max){
                        max=valor
                        pair=c(clust1,clust2)
                        }
                    }
                }
            }
        }

for ( i in individuals[individuals<0]){
    i=-i
    for ( j in individuals[individuals<(-i)]){
        j=-j
        if(a[i,j]>=max){
            max=a[i,j]
            pair=c(-i,-j)
            }
        }
    }

if (is.na(A[1] )){
    A<-matrix(pair,ncol=2,byrow=TRUE)
    B<-c(max)
    }
else{
    A<-rbind(A, c(pair[1],pair[2]))
    B<-c(B,max)
    }
```

```
    clusters_indices<-c(clusters_indices, length(clusters_indices)+1)

    x1=pair[1]
    x2=pair[2]
    indice=length(clusters)+1
    if (x1>0){
        clusters_indices[x1]<-0
        x1=c(clusters[[x1]])
        }
    else{
        individuals[-x1]<-0
        }
    if (x2>0){
        clusters_indices[x2]<-0
        x2=c(clusters[[x2]])
        }
    else{
        individuals[-x2]<-0
        }

    clusters[[indice]]<-c(x1,x2)
    contador=contador+1

    }
C<-list()
C$merge<-A

C$height<-2*sqrt(1-B)
C$order<- 1:length(b)
C$labels<- b
C$method<- "ASI"
C$dist <- "Lerman"
class(C)<-"hclust"
C$order<-order.dendrogram(as.dendrogram(C))
C
}
```

A.2. Conjunto de datos Animaux

El archivo animaux.csv contiene un conjunto de datos formado por 41 animáles y 75 variables binarias que representan la presencia o no de características en los animales. El conjunto de datos es el de la tabla A.1

	Affectueux	Agile	Agressif	Angoissant	Attirant	Beau	Bete	Blanc	Bon	Bruyant	Calme	...
Aigle	0	1	0	1	1	1	0	0	0	0	1	
Ane	0	0	0	0	0	1	1	0	0	0	1	
Autruche	0	0	1	1	0	0	1	0	1	0	0	
Baleine	0	0	0	1	1	1	0	1	0	1	1	
Bouc	0	1	1	0	1	0	1	0	0	1	0	
Canard	0	0	1	0	0	1	1	0	1	1	1	
Chamois	0	1	0	0	0	1	0	0	0	0	0	
Chat	1	1	0	0	1	1	1	0	0	0	0	
Chien	1	0	0	0	0	0	1	0	0	1	0	
Cigale	0	0	0	0	0	1	0	0	0	1	0	
Corbeau	0	0	1	1	0	1	0	0	0	0	0	
Couleuvre	0	0	0	1	0	0	0	0	0	0	0	
Crocodile	0	0	1	1	0	0	0	0	0	0	1	
Crotale	0	1	1	1	0	0	0	0	0	1	0	
Dauphin	0	0	0	0	1	1	0	0	0	0	1	
Fourmi	0	0	0	0	1	0	0	0	0	0	0	
Grenouille	0	1	0	0	0	1	1	0	1	1	0	
Guepe	0	1	1	1	0	0	0	0	0	1	0	
Lapin	0	0	0	0	0	0	1	1	1	0	1	
Lion	0	0	1	0	1	1	0	0	1	0	0	
Loup	0	0	1	1	0	1	0	1	0	1	0	
Lynx	0	1	1	0	0	1	0	0	0	0	0	
Mouche	0	0	0	0	0	0	1	0	0	1	0	
Mouton	1	0	1	0	0	0	1	1	0	1	1	
Mygale	0	0	1	1	0	0	0	0	0	0	0	
Ours	1	0	0	1	1	1	0	0	0	0	0	
Pieuvre	0	1	1	1	0	0	0	0	1	0	0	
Pigeon	0	0	0	0	0	0	1	1	0	0	0	
Porc	0	0	1	0	0	0	1	0	1	1	0	
Poule	0	0	0	0	0	0	1	0	1	1	1	
Rat	0	0	1	1	0	0	1	0	0	0	0	
Renard	0	0	0	0	0	1	0	0	0	0	0	
Requin	0	0	1	1	0	1	0	1	0	0	0	
Singe	1	1	0	0	0	0	1	0	0	1	0	
Taureau	0	0	1	0	0	0	1	0	0	0	1	
Tigre	0	1	1	1	1	1	0	1	0	0	1	
Tortue	0	0	0	0	0	0	0	0	0	0	1	
Vache	0	0	0	0	0	0	1	1	1	0	1	
Vautour	0	1	1	1	0	1	0	0	0	0	1	
Vison	0	0	0	1	0	1	0	0	0	0	0	
Zebre	0	0	0	0	0	1	1	1	0	0	1	

Tabla A.1: Conjunto de datos Animaux