Técnicas de aprendizaje automático en estudios epidemiológicos longitudinales. Aplicación a la Cohorte EpiChron de investigación en multimorbilidad.

Elena Campo León

Trabajo de fin de Máster

Máster Universitario en Modelización e Investigación Matemática, Estadística y Computación

Universidad de Zaragoza

Director del trabajo: José Tomás Alcalá Nalvaiz
Antonio Gimeno Miguel

Prólogo

A lo largo del presente trabajo se presentan algunas técnicas de Machine Learning, aplicables a estudios longitudinales de tipo cohorte, en particular aplicables a estudios de investigación de tipo médico. El objetivo final será la aplicación de estas técnicas para construir trayectorias de multimorbilidad entre enfermedades que aparecen con un determinado patrón frecuente, es decir, determinar la secuencia temporal más probable en el diagnóstico.

Se estudia el clustering difuso, técnica de aprendizaje no supervisado que permite realizar clusters o agrupamientos de individuos u objetos con características similares. Para ello se hace uso de distancias a los centros de cada grupo, considerando individuos similares aquellos que tienen distancia mínima al centro de dicho grupo. Difiere del clustering clásico en que los individuos tienen una probabilidad de pertenencia a cada clúster. En nuestro estudio se utiliza el clúster difuso con el objetivo de agrupar las diferentes enfermedades con las que trabajamos.

Las reglas de asociación ponderadas, nos permiten descubrir relaciones ocultas en grandes cantidades de datos. De esta forma la presencia de un determinado item (artículo, enfermedad...) implica la presencia de otro u otros con mayor o menor probabilidad. El hecho de recurrir a la ponderación permite asignar unos pesos a unos items determinados por el motivo que sea, por ejemplo que la venta de un determinado producto en unas pocas unidades provoque muchos más beneficios que la venta de otro en decenas o cientos de unidades. Aplicándolo a nuestro estudio longitudinal, nos permite determinar relaciones existentes entre diferentes enfermedades y establecer los patrones que aparecen con mayor frecuencia.

La puesta en práctica de estas técnicas se realiza sobre un conjunto anonimizado de datos de pacientes de entre 65-75 años extraídos de la Cohorte EpiChron, de investigación en multimorbilidad, que contiene datos sociodemográficos, clínico-farmacológicos y de uso de servicios sanitarios para la población adscrita al sistema de salud público de Aragón. Se dispondrán de datos sobre aparición de enfermedades en pacientes durante los años 2010-2019. Como se ha mencionado, el objetivo principal será obtener y modelar la evolución de trayectorias de multimorbilidad, basadas en grupos. De esta forma, el conocimiento de la evolución (más probable) de un paciente que padece inicialmente una enfermedad ubicada en un determinado clúster o grupo, permite enfocar las políticas de Salud Pública a una mejor prevención y gestión de recursos.

En relación al trabajo con la base de datos, inicialmente se realiza un procesado de los datos agrupando y eliminando las enfermedades cuya prevalencia en el año 2010 sea inferior al 1 %. Mediante el clúster difuso, se asigna para cada enfermedad una probabilidad de pertenencia a cada uno de los clusters establecidos. Las reglas de asociación ponderadas nos permiten establecer los distintos grados de asociación entre los enfermedades, y generar trayectorias de multimorbilidad asociadas a los patrones elegidos, mediante el estudio de sus rangos medios. Para estudiar la variabilidad de dichos rangos se emplea una técnica de remuestreo Bootstrap. Por último, se generarán grafos que muestren el flujo entre enfermedades de forma que se aprecia la flexibilidad de las trayectorias obtenidas.

Summary

The aim of this study is to present some techniques of Machine Learning, wich are suitable to Longitudinal studies of the cohort type. Our goal will be the application of these techniques to build multimorbidity trajectories between diseases that appear with a certain frequent pattern, it means, determine the temporal sequence most likely at medical diagnosis.

Here, it is used fuzzy c-means algorithm to identify multimorbidity cluster. The algorithm, which belongs to the family of soft clustering algorithms, estimates c cluster centers (similar to k-means algorithm) but with fuzziness so that individuals may belong to more than one cluster. The use of a fuzzy cluster analysis over a hard cluster analysis helps to better handle the stochastic nature of some disease association, the potential noise stemming from the measurement, and the variance due to between-individual differences. This technique, allows obtain clusters of diseases and a membership matrix which indicates the degree of participation of each patology in each cluster.

Association rule mining is a method to uncover the combinations of items that occur together frequently, discovering interesting relations. It is often applied in the studies of supermarket sales and customer behaviors and is gaining attention in the clinical research. Weighted Association Rule Mining are use to assign weights to certain items, for example, depending the product, the sale of some units is not the same as hundreds of units. Applying it to our longitudinal study, it allows us to determine existing relationships between different diseases and establish the patterns that appear with more frequency.

The implementation of these techniques is carried out on a data set of patients between 65-75 years extracted from the EpiChron database, which contains sociodemographic, clinical- pharmacological and use of medical services for the entire Aragonese population attached to the system of public health. Data will be available on the occurrence of diseases in patients during the years 2010-2019. As mentioned, the main objective will be to obtain and model the evolution of group-based multimorbidity patterns. In this way, knowledge of evolution of a patient who initially suffers from a disease located in a certain cluster or group, makes it possible to focus Public Health policies on better prevention and drug management.

Related to the database, diseases whose prevalence in 2010 is less than 1 % are eliminated. The probability of belonging to each of the clusters is assigned for each disease, With fuzzy cluster. The weighted association rules allow us to establish the different degrees of association between diseases, and generate multimorbidity secquences studying their average ranks. In order to study the variability of these ranges, a Bootstrap resampling technique is employed. Finally, associated graphs to each pattern will show us the flexibility of the trajectories obtained.

Índice general

Capítulo 1

Clúster Difuso

El objetivo principal de la mayoría de métodos de clustering es dar información útil mediante el agrupamiento (no etiquetado) de los datos en clusters o grupos con características similares. La similitud entre individuos se define a través de la distancia entre cada individuo y el centro del clúster.La función de distancias suele ser la norma euclídea, que genera agrupamientos circulares, pero se puede extender en caso de querer mayor robustez, a la norma de Mahalanobis y en este caso los elementos asociados a cada clúster forman elipses. Así mismo existen normas para grupos rectangulares, cilíndricos, etc.

La partición en clusters generada sobre el conjunto de datos puede ser de dos tipos:

- **Hard Clustering**: donde cada individuo del conjunto pertenece a un único cluster. Un ejemplo es el algoritmo K-means, [5].
- **Soft Clustering**: a cada individuo se le asigna una credibilidad o probabilidad de pertenencia a cada clúster. Puede decirse que la frontera que generan estos algoritmos es blanda o difusa. El algoritmo más usado es el Fuzzy C-means.

A grandes rasgos puede decirse que el algoritmo Fuzzy C-means consta de dos procesos. Inicialmente, para cada número de clusters, que varía en un intervalo prefijado, se obtienen las probabilidades de asignación de cada individuo a cada clúster. En el segundo paso se evalúan unos índices de validación que permiten decidir cuál es el número óptimo de clusters y por tanto que partición de la primera fase es la óptima.

1.1. Algoritmo Fuzzy C-Means

El algoritmo FCM, [11]fue propuesto por Bezdek en 1973. Desde entonces han ido variando las normas usadas en las distancias así como la determinación de los distintos centroides en los clusters (Dunn, 1974; Dave y Bhaswan, 1992; Krishnapuram et al, 1992; Man y Gath, 1994) manteniendo su esencia[2],[11]. Vamos a describirlo basándonos en la definición introducida por Bezdek.

Sea $X = \{x_1, ..., x_n\}$ un conjunto de n individuos, donde $x_k \in R^p$, siendo p el número de variables observadas. Sea c el parámetro que indica el número de clusters. Se dice que el conjunto de clusters $P = \{C_1, ...C_c\}$ forman una partición difusa de X si y solo si se cumple:

i) $0 \leq u_{ik} \leq 1, \quad \forall \quad i = 1, ..., c, \quad k = 1, ..., n$

ii) $\sum_{i=1}^{c} u_{ik} = 1, \quad \forall \quad k = 1, ..., n$

iii) $0 \leq \sum_{k=1}^{n} u_{ik} \leq 1, \quad \forall \quad i = 1, ..., c$

siendo u_{ik} el grado de pertenencia al clúster i del individuo k.

Fijado el número de clusters c, se trata de buscar la mejor partición difusa, para ello se minimiza la suma ponderada de errores al cuadrado en cada grupo. Se define como la siguiente función objetivo:

$$J_m(U,V;X) = \sum_{k=1}^{n}\sum_{i=1}^{c} u_{ik}^m \|x_k - v_i\|_A^2, \quad 1 < m < \infty, \tag{1.1}$$

donde $V = \{v_1, ..., v_c\}$ es el vector de los centros de los clusters, inicialmente desconocidos. $U = [u_{ik}]$ es la matriz de pertenencias. La matriz A es la asociada a la norma elegida, m es el factor difuso, $m \geq 1$, observar que la partición se vuelve más difusa cuanto mayor es el valor de m y dejaría de serlo en caso de $m = 1$.

Bezdek demostró en 1981, que si $\|x_k - v_i\|_A > 0 \quad \forall i,k$, entonces la partición asociada al par (U,V) minimiza J_m si el factor difuso $m > 1$ y además,

$$v_i = \frac{\sum_{k=1}^{n} u_{ik}^m X_k}{\sum_{k=1}^{n} u_{ik}^m} \quad \forall \quad 1 \leq i \leq c \tag{1.2}$$

$$u_{ik} = \frac{1}{\sum_{j=1}^{c} \left(\frac{\|x_k - v_i\|_A^2}{\|x_k - v_j\|_A^2}\right)^{\frac{1}{m-1}}} \quad \forall \quad 1 \leq i \leq c, 1 \leq k \leq n \tag{1.3}$$

Entre otros, J_m se puede minimizar mediante el método iterativo de Picard, [2], el esquema se muestra en el algoritmo 1:

Algorithm 1 Algoritmo Fuzzy C-Means.

-Se fija el número de clusters, $1 < c < n$.
-Se elige el factor de difusión $m > 1$.
-Se elige la norma/distancia.
-Se elige la tolerancia ε que permita establecer el criterio de parada.
-Se genera la matriz de pertenencias inicial U de forma aleatoria.
while $\|U^{(p)} - U^{(p+1)}\| > \varepsilon$, *donde p es la iteracion.* **do**

1. Calcular nuevos centros V en cada clúster aplicando (1.2)
2. Calcular las distancias $d_{ik} = \|x_k - v_i\|_A$ entre el individuo k y el centro $v_i \in V$ del cluster i.
3. Actualizar la matriz de pertenencia aplicando (1.3).
4. Comprobar la condición de parada

end

1.2. Índices de validación para el algoritmo Fuzzy C-means

El objetivo de los índices de validación, aplicados sobre las particiones generadas por el algoritmo FCM, es determinar el número óptimo de clusters.

Considerar c_{min}, c_{max} respectivamente, el mínimo y máximo número de clusters establecido, entonces para cada $c \in [c_{min}, c_{max}]$ se aplica FCM para generar una partición y sobre ella se calcula el índice de validación. Un criterio habitual de elección de número de clusters óptimo, es tomar el asociado a la

partición que hace mínimo el valor del índice.

Existe una amplia variedad de índices de validación,[12], entre los más populares destacan, el coeficiente de partición V_{PC} o la entropía de partición V_{PE} propuestos por Bezdek (1974 y 1975), el coeficiente de separación de Gunderson (1978), el exponente de proporción o el funcional de datos uniforme de Windham (1981 y 1982). Otros autores como Libert y Roubens(1983), Windham et al.(1989), Fukuyama y Sugeno(1989), Xie y Beni (1991); Gindy et al (1995), han propuesto asímismo otros índices de validación, algunos de los cuales se encuentran definidos en el cuadro 1.1.

Índice de validación	Funcional	Nº clusters óptimo
Coeficiente de partición	$V_{PC}(U) = \frac{1}{n}(\sum_{k=1}^{n}\sum_{i=1}^{c} u_{ik}^2)$	Max$V_{PC}(U,c_i,m)$
Entropía de partición	$V_{PE}(U) = -\frac{1}{n}(\sum_{k=1}^{n}\sum_{i=1}^{c} u_{ik} log_a(u_{ik}))$	Min$V_{PE}(U,c_i,m)$
Fukuyama y Sugeno	$V_{FS,m}(U,V;X) = \sum_{i=1}^{c}\sum_{k=1}^{n} u_{ik}^m(\|x_k - v_i\|^2 - \|v_i - \bar{v}\|_A^2)$	Min$V_{FS,m}(U,c_i,m)$
Xie y Beni	$V_{XB}(U,V;X) = \frac{\sum_{i=1}^{c}\sum_{k=1}^{n} u_{ik}^m \|x_k - v_i\|^2}{n(min\{v_i - v_j\})}$	Min$V_{XB}(U,c_i,m)$

Cuadro 1.1: Índices de validación

Si establecemos una breve comparativa, los coeficientes de partición $V_{PC}(U)$ y entropía de partición $V_{PE}(U)$ usan únicamente los términos u_{ik} de la matriz de pertenencias U. Diversos estudios realizados (Pal y Bezdek, 1995) muestran que maximizar V_{PC} (minimizar V_{PE}) a menudo da lugar a buenas interpretaciones sobre el conjunto de datos, sin embargo presentan el inconveniente de que al ser una función sólo del conjunto de datos, se pierden algunas propiedades geométricas relacionadas con la distancia a los centros de cada clúster, esta propiedad la introduce el coeficiente de separación de Gunderson, en 1978. Asímismo es utilizada por el índice de Fukuyama y Sugeno y por el índice de Xie y Beni, (Pal y Bezdek, 1995). Además se muestra la idoneidad de introducir en estos índices el factor difuso m.

Un ejemplo de búsqueda de criterios que permitan determinar el número óptimo de clusters, lo podemos encontrar en M. Ramze Rezaee et al (1998) [11], donde el índice V_{CWB} (Compose Within and Between scattering), construido establece un equilibrio entre compactitud y separación, pues la partición óptima debe ser aquella en la que los clusters están más alejados unos de otros y además son compactos en cuanto a similaridad en sus componentes. Se define el índice para cada c-partición y tomando la norma euclídea como

$$V_{CWB}(U,V) = \alpha Scat(c) + Dis(c)$$

donde el funcional de la distancia es

$$Dis(c) = \frac{Max\|v_i - v_j\|}{Max\|v_i - v_j\|}\sum_{k=1}^{c}(\sum_{z=1}^{c}\|v_k - v_z\|)^{-1} \quad \forall i,j \in \{1,...,c\}$$

$\alpha = Dis(c_{max})$ y la dispersión media para c-clusters

$$Scat(c) = \frac{\frac{1}{c}\sum_{i=1}^{c}\|\sigma(v_i)\|}{\|\sigma(X)\|}$$

siendo

$$\sigma(v_i)^p = \frac{1}{n}\sum_{k=1}^{n} u_{ik}(x_k^p - v_i^p)^2$$

la p-ésima componente de la varianza difusa del cluster i y

$$\sigma(X)^p = \frac{1}{n}\sum_{k=1}^{n}(x_k^p - \bar{x}^p)^2$$

la p-ésima componente de la varianza total del conjunto de individuos.

El término $Scat(c)$ permite controlar la compacidad media de los clusters, cuanto mayor es su valor mayor dispersión.

El término $Dis(c)$ indica el grado de separación entre clusters, su valor aumenta conforme hay más clusters. Esta muy determinado por las propiedades geométricas de los centros de los clusters.

El parámetro ponderador α permite equilibrar el peso de cada termino puesto que el rango de magnitud es distinto en ambos.

El número óptimo de clusters se considera aquel que minimiza el índice V_{CWB}.

Pese a su complejidad, el interés de este índice radica, como sugieren M. Ramze Rezaee et al (1998), en que frente a los otros presentados en el cuadro 1.1 es el que presenta mayor fiabilidad, pues en todas las muestras escogidas, se obtuvo el mismo número de clusters óptimo.

Diversos estudios (M.Rawashdeh y A. Ralescu, 2012) [12] muestran la idoneidad de usar una generalización del índice silueta, frente a otros índices mencionados. Originariamente, Rousseeuw implementó este índice Crisp Silueta (CS) en 1987, para generar particiones de hard-clustering. Posteriormente, en 2006, Campello y Hruschka, [16], [2], definen el índice Silueta Fuzzy, como una generalización del índice CS, para hard-clustering, adaptado al caso de fuzzy-clustering.

El índice silueta fuzzy se define como:

$$FS = \frac{\sum_{j=1}^{n}(u_{pj} - u_{qj})^{\alpha} \cdot s_j}{\sum_{j=1}^{n}(u_{pj} - u_{qj})^{\alpha}}$$

donde

$$s_j = \frac{b_{pj} - a_{pj}}{max\{a_{pj}, b_{pj}\}}$$

es la silueta asociada al individuo j, siendo a_{pj} la distancia media del individuo j al resto de elementos del clúster p. b_{pj} es la distancia mínima del individuo j al resto de clusters distintos de p.

Los elementos u_{pj}, u_{qj} se refieren para cada columna j, a los dos mayores elementos, de la matriz de pertenencia U (una vez aplicado FCM para un número fijo de clusters), es decir para cada individuo se eligen los dos clusters con mayor probabilidad de pertenencia.

Observar que $\alpha \geq 0$ es un parémetro de ponderación opcional (por defecto es 1) sobre los términos del sumatorio, cuanto más próximo esta a 0, el valor de FS, se aproxima al de Crisp Silueta (CS), de igual forma CS se obtiene como un caso particular de FS cuando $\alpha = 0$. Al aumentar el valor de α permite la detección de subclusters en áreas con una densidad muy alta de datos o datos contaminados con ruido. Puede verse por lo tanto como una herramienta exploratoria para analizar el conjunto de datos.

1.3. Algoritmo de validación Fuzzy C-means

Como se ha mencionado anteriormente, para elegir el número óptimo de clusters, c_{opt},se debe evaluar el índice de validación elegido dentro del rango de clusters $[c_{min}, c_{max}]$ que generan las posibles particiones del conjunto de datos. El esquema general del algoritmo que permite decidir dicho número

óptimo es el algoritmo 2.

Algorithm 2 Algoritmo de validación Fuzzy C-means.

-Inicialización $c_{opt} = c \leftarrow c_{max}$
-Inicialización de índice de validación $V(c) \leftarrow V(c_{max})$
for $c \in [c_{min}, c_{max}]$ **do**
  while $\|u_{ik}(p) - u_{ik}(p-1)\| > \varepsilon$ **do**
    -Aplicar FCM $\rightarrow$ se actualizan centros de clusters v_i y factores de pertenencia u_{ik}
    -Calcular $V(c)$
    end
    if $V(c^{(t)}) < V(c^{(t-1)}) \rightarrow$ **then**
    $c_{opt} \leftarrow c^{(t)}$
    end
  end

1.4. Paquetes en R: Fclust

El algoritmo FCM esta implementado en diferentes funciones de diversos paquetes en R: *cluster* (Maechler et al.,2017), *clue* (Hornik,2005), *e1071* (Meyer et al.,2017), *skmeans* (Hornik et al.,2012), *vegclust*(De Caceres et al.,2010), *ppclust* (Cebeci et al.,2018) y *fclust* (Ferraro y Giordani,2015).

Entre ellos destacar *fclust* [14], [20], que ofrece herramientas específicas para particionamiento de datos como algoritmos de clustering fuzzy, computación de índices de validación y diversos gráficos para la visualización de los resultados de efectuar el clustering. La versión actual (2.1.1) ofrece múltiples mejoras,respecto de otras anteriores, entre ellas destacan la selección automática del número óptimo de clusters, especificando el índice que se desea usar entre los disponibles en el paquete y que se pueden consultar en cuadro 1.2. Por defecto se usa el índice silueta fuzzy, "SIL.F".

Índice de validación	Argumento en R
Coeficiente de partición (Bezdek)	PC
Coeficiente de partición modificado	MPC
Entropía de partición (Bezdek)	PE
Entropía de partición(Xie y Beni)	XB
Silueta (crisp)	SIL
Silueta (fuzzy)	SIL.F

Cuadro 1.2: Lista de índices de validación para cluster fuzzy disponibles en el paquete *fclust*

Destacar la función NEFRC, que permite aplicar clustering fuzzy sobre datos relacionales y para una métrica no euclídea, pudiendo establecer la matriz de distancias en función de la métrica que más se adapte a los datos:

```
NEFRC(D, k, m, RS, startU, index, alpha, conv, maxit, seed)
```

Sus argumentos son los siguientes:

- D, es el conjunto de datos o matriz de distancias.
- k, indica el número o vector de clusters para el que se calculará el INDEX, con las posibilidades especificadas en cuadro 1.2.

- m, indica el parámetro de difusión.
- RS, es el número de inicios aleatorios.
- startU, es un inicio racional para la matriz de pertenecias U.
- conv, criterio de convergencia.
- index, representados en el cuadro 1.2.
- alpha, coeficiente de ponderación para el índice silueta fuzzy.
- maxit, máximo numero de iteraciones.
- seed, semilla para generación de números aleatorios.

Como se ha mencionado el objetivo de usar clúster difuso es establecer un grado de pertenencia a cada una de las patologías que se estudian. Previamente a ello, se realiza una partición de tipo jerárquico, perteneciente a técnicas de tipo hard-clustering. De esta forma se realiza una estimación del número de clusters óptimo. Mediante el clúster fuzzy y más concretamente a través de la función NEFRC aplicada sobre la matriz de distancias y para el número de cluster aconsejado por el hard-clustering, se obtienen la matriz de pertenencias y la adjudicación a cada clúster, tomando cómo criterio que la pertenencia de una enfermedad a un clúster, sea superior al 50 %. La matriz de distancias, se obtiene con la métrica de "Kulczynski2" adecuada para datos de tipo binario, como es nuestro caso.

Capítulo 2

Reglas de asociación

2.1. Fundamentación teórica de las reglas de asociación

El análisis de asociación, [17] permite descubrir relaciones ocultas en grandes cantidades de datos. Dichas relaciones se presentan mediante reglas de asociación.

El estudio de las reglas de asociación se introduce originariamente con el *problema de la cesta de la compra*, donde se trata de detectar cuándo la ocurrencia (compra) de un artículo esta asociada a la ocurrencia (compra) de otros artículos en la la misma transacción (compra de productos en el ejemplo), es decir dado un conjunto de transacciones se trata de encontrar reglas que describen tendencias en los datos. El estudio de las reglas de asociación permiten realizar promociones y ofertas, y desarrollar una política de inventario y relación con el cliente más eficiente de caras a optimizar las ventas.

Tomamos como referencia el ejemplo de *la cesta de la compra*, en el cuadro 2.1 se muestran en cada fila las transacciones, es decir, la compra realizada por cada cliente. Cada fila tiene un identificador TID.

TID	itemset
1	Pan,leche,huevos
2	Pan, pañales,cerveza
3	Leche,pañales,cerveza
4	Pan,leche,pañales,cerveza
5	Leche,huevos,cerveza

Cuadro 2.1: Transacciones.

Un ejemplo de regla de asociación fácilmente extraíble de la tabla anterior, podría ser *Pañales*$\longrightarrow$*Cerveza*, es decir una persona que compre pañales muy probablemente compre también cerveza. La política del supermercado, sería colocar los productos que los clientes compran conjuntamente, en estanterías próximas.

El uso de reglas de asociación se aplica en diversos campos como la bioinformatica, diagnóstico médico, web mining o análisis de datos en todas las áreas.

2.1.1. Definiciones previas

Sean dos itemsets o transacciones, X e Y, pertenecientes a una matriz de transacciones y sea la regla de asociación asociada a dichos itemsets $X \longrightarrow Y$. Veamos algunas definiciones asociadas a estos con-

ceptos:

- **k-Itemset**: es un conjunto con k items(artículos), donde $k = 1, 2, ...$
- **Soporte de un intemset**: es la fracción de las transacciones que contienen un itemset.
- **itemset frecuente**: es un itemset con un soporte igual o superior a un umbral de soporte establecido por es usuario.
- **Regla de asociación**: es una expresión de la forma $X \longrightarrow Y$, donde X, Y son itemsets.

Las medidas de evaluación de las reglas de asociación son:

- **Soporte de la regla**: se denota por $supp(X \longrightarrow Y)$ es la fracción de las transacciones que contiene tanto a X como a Y, es decir, $supp(X \cup Y)$.
- **Confianza de la regla**: se denota por $conf(X \longrightarrow Y)$ es la fracción de las transacciones en las que aparece X, que también aparece Y, es decir, la confianza mide con que frecuencia aparece Y en las transacciones que incluyen X.
- **Lift**: es una medida que cuantifica la relación entre los itemsets X e Y. Indica la proporción entre el soporte observado de un conjunto de items respecto de su soporte teórico bajo el supuesto de independencia. Dicho de otro modo, compara la frecuencia de un patrón observado con respecto a lo que se esperaría ver ese patrón solo por azar. Se define como

$$Lift(X \longrightarrow Y) = \frac{Supp(X \longrightarrow Y)}{Supp(X)Supp(Y)}$$

Siguiendo con el ejemplo anterior, se puede interpretar que en la tabla 2.1

$$supp(\{pañales\}) = \frac{3}{5}$$

y

$$supp(\{cerveza\}) = \frac{4}{5}$$

y en el caso de la regla de asociación,

$$supp(\{cerveza\} \longrightarrow \{pañales\}) = \frac{3}{5}$$

A su vez la confianza de la regla de asociación anterior se calcula

$$conf(\{cerveza\} \longrightarrow \{pañales\}) = \frac{supp(\{cerveza\} \longrightarrow \{pañales\})}{supp(\{cerveza\})} = \frac{3}{4}$$

y para el caso del Lift asociado a la regla,

$$Lift(\{cerveza\} \longrightarrow \{pañales\}) = \frac{supp(\{cerveza\} \longrightarrow \{pañales\})}{supp(\{cerveza\})supp(\{pañales\})} = \frac{1}{4}$$

2.1.2. Formulación del problema

Sea $I = \{i_1, i_2, ..., i_d\}$ el conjunto de todos los items y sea $T = \{t_1, ..., t_N\}$ el conjunto de todas las transacciones, cada una formada por elementos de I.

Sean X e Y subconjuntos o itemsets disjuntos con elementos en I. Se define $\sigma(X) = |\{t_i | X \subseteq t_i, t_i \in T\}|$, como el número de transacciones donde aparece el itemset X. Entonces, dado un conjunto de transacciones T, se trata de encontrar todas las reglas de asociación, tal que:

- el soporte de la regla de asociación, $Supp(X \longrightarrow Y) = \frac{\sigma(X \cup Y)}{N}$ sea mayor o igual que el umbral mínimo de soporte, *MinSupp*. Es decir, $Supp(X \longrightarrow Y) \geq MinSupp$.
- la confianza de la regla de asociación, $Conf(X \longrightarrow Y) = \frac{\sigma(X \cup Y)}{\sigma(X)}$ sea mayor o igual que el umbral mínimo de confianza, *MinConf* . Es decir, $Conf(X \longrightarrow Y) \geq MinConf$

Una solución a este planteamiento sería enumerar todas las reglas de asociación posibles, calcular el soporte y confianza de cada regla y eliminar las reglas que no superen los umbrales de soporte y confianza, sin embargo es computacionalmente prohibitivo, ya que el número de reglas posibles asciende a $3^d - 2^{d+1} + 1$, siendo d el número de items, como se puede consultar en capítulo 6 de [17].

Para resolver este problema, la solución propuesta por la mayoría de algoritmos, es descomponer el problema en dos etapas:

1. **Generación de los itemsets frecuentes**: identificar los itemsets con soporte mayor o igual que el *MinSupp*, esos se consideraran frecuentes.
2. **Generación de reglas de asociación**: Obtener reglas de asociación con una confianza elevada a partir de cada itemset frecuente, donde cada regla es una partición binaria del itemset.

La generación de los itemsets frecuentes sigue siendo un proceso computacionalmente costoso, ya que un conjunto de datos que contiene d items puede generar $2^d - 1$ itemsets frecuentes, excluyendo el conjunto nulo.

Para reducir la complejidad computacional de la generación de itemsets frecuentes se proponen diversos métodos:

1. Reducir el número de candidatos, mediante el uso de técnicas de poda. Algunos algoritmos de ejemplo son Algoritmo Apriori y DHP (Direct Hashing and Pruning).
2. Reducir el número de transacciones, conforme aumenta el tamaño del itemset. Ejemplo de ello es el algoritmo Apriori TID.
3. Reducir el número de comparaciones, mediante el uso de estructuras de datos eficientes para almacenar los candidatos o transacciones de forma que no haya que comparar cada candidato con todas las transacciones. Por ejemplo los FP-tree, estructura base de los algoritmos FP-Growth.

Veamos en que consiste en particular el algoritmo Apriori, que permite reducir la complejidad computacional en la generación de itemsets frecuentes.

2.1.3. Principio Apriori

El principio Apriori permite reducir el número de candidatos a itemset frecuente, a través de la medida de soporte siguiendo el siguiente principio:

Teorema 1. *(Principio Apriori) Si un itemset es frecuente, entonces, también lo son todos sus subconjuntos*

Ejemplo: Considerar los items $\{a,b,c,d\}$ que generan todos los itemsets posibles de la figura 2.1. Del principio Apriori se deduce que si el itemset $\{cde\}$ es frecuente entonces todos los itemsets que contiene, son frecuentes, son los sombreados en 2.1. Análogamente, si un itemset $\{ab\}$ no es frecuente, es decir, tiene soporte inferior al umbral, entonces todos los supersets que lo contienen son no frecuentes. En consecuencia, como se muestra en la figura 2.2 el subgrafo asociado se puede eliminar. Esta estrategia se conoce con el nombre de **poda basada en el soporte** y se basa en la **propiedad de antimonotonía**, es decir, si un itemset es frecuente, lo son todos los subconjuntos contenidos en el.

Definicion 1. *(Propiedad de Monotonía) Sea I un conjunto de items y sea $J = 2^I$ el número de subconjuntos. Sea Supp el soporte, entonces:*

$$\forall X,Y \in J : (X \subseteq Y) \longrightarrow Supp(X) \leq Supp(Y)$$

Por otro lado, se tiene:

Definicion 2. *(Propiedad de Anti-monotonía) Sea I un conjunto de items y sea $J = 2^I$ el número de subconjuntos. Sea Supp el soporte, entonces:*

$$\forall X,Y \in J : (X \subseteq Y) \longrightarrow Supp(X) \geq Supp(Y)$$

En consecuencia, la propiedad de antimonotonia para el soporte permite aplicar la estrategia de la poda en el algoritmo Apriori

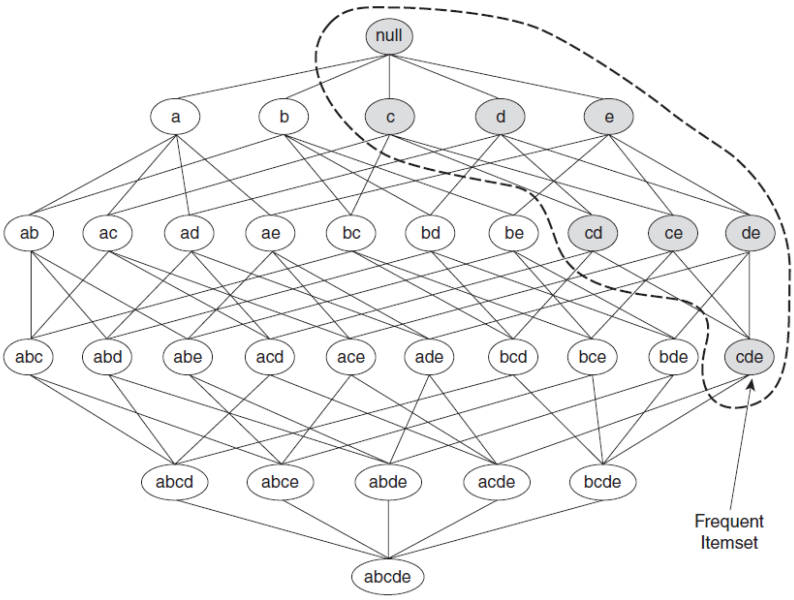


Figura 2.1: Itemsets frecuentes aplicando principio Apriori

2.1.4. Generación del itemset frecuente con el algoritmo Apriori

La implementación del algoritmo Apriori como técnica de poda, basada en el soporte, para controlar el crecimiento exponencial de los itemsets candidatos, fue pionera, propuesta en 1994 por Agrawal [1].

El pseudocódigo para generar los itemsets frecuentes se muestra en el Algoritmo 1. Sea C_k el conjunto de k-itmsets candidatos y sea F_k el conjunto de k-itemsets frecuentes:

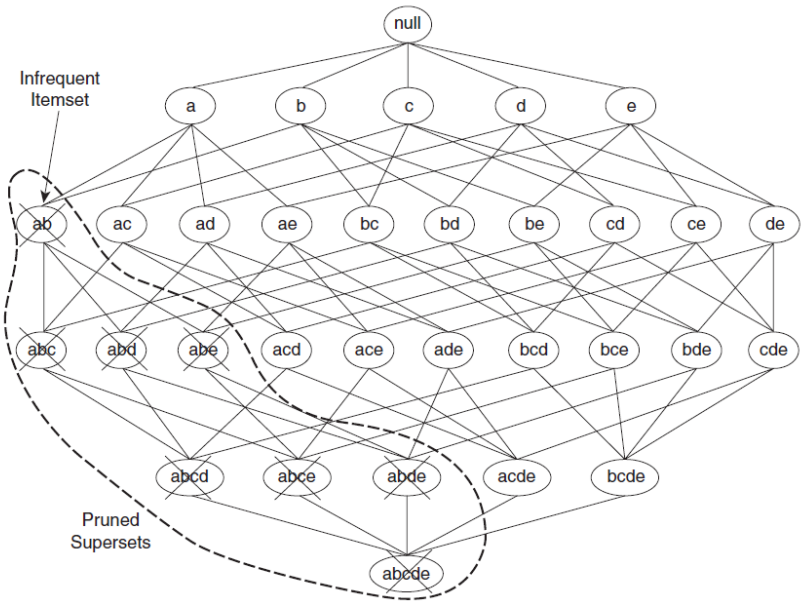


Figura 2.2: Estrategia de poda de itemsets no frecuentes en algoritmo Apriori

1. El algoritmo inicialmente, determina el soporte de cada item. Una vez completado este paso, se conocen los 1-itemsets frecuentes, F_1.

2. A continuación el algoritmo genera iterativamente nuevos candidatos k-itemsets usando los k-1 itemsets frecuentes encontrados en las anteriores iteraciones, mediante la función *apriori-gen*.

3. Para evaluar el soporte de cada candidato k-itemset, se usa la función subset. Esta determina el número de transacciones en las que aparece cada k-itemset.

4. Una vez determinados los soportes de cada k-itemset, se eliminan mediante la estrategia de poda basada en el umbral mínimo del soporte, aquellos que no superen el *MinSupp*. La función soporte se denotará por σ.

5. El algoritmo finaliza cuando no se generan nuevos itemsets frecuentes $F_k = \emptyset$

Un esquema del algoritmo Apriori para generar itemsets frecuentes particularizado al problema de la cesta de la compra, sería el que se muestra en la figura 2.3. En tono rosa sombreado, se observan los itemsets que son eliminados en cada paso debido a que no superan el umbral del soporte, en este caso $MinSupp = 3$.

MinSupp=3

Candidatos 1-itemsets	
1-Item	Support
Pan	4
Leche	4
Huevos	2
Cerveza	4
Pañales	3

Candidatos 2-itemsets	
2-Itemsets	Support
Pan,Leche	3
Pan, Cerveza	3
Pan,Pañales	2
Leche, Pañales	2
Cerveza, Pañales	3
Leche, Cerveza	3

Candidatos 3-itemsets	
3-Itemsets	Support
Pan,Leche,Cerveza	2
Cerveza, Pañales, Leche	2

Figura 2.3: Generación de itemsets frecuentes con algoritmo Apriori

Algorithm 3 Generación de itemsets frecuentes con el algoritmo Apriori.

k=1.
$F_k = \{i | i \in I \wedge \sigma(i) \geq N \times minSupp\}$ (Encontrar todos los 1-itemsets frecuentes).
for *k hasta que* $F_k = \emptyset$ **do**
  $C_k = apriori - gen(F_{k-1})$ (Generar itemsets candidatos)
  for $t \in T$ **do**
    $C_t = subset(C_k, t)$ (Identificar todos los candidatos que pertenecen a transaccion t)
    for $c \in C_t$ **do**
      $\sigma(c) = \sigma(c) + 1$ (contabilizar el soporte de cada candidato)
    end
  end
  $F_k = \{c | c \in I \wedge \sigma(c) \geq N \times minSupp\}$ (Extraer los k-itemsets frecuentes, eliminando los no frecuentes)
end
Resultado=$\cup F_k$

2.1.5. Generación de itemsets frecuentes con el algoritmo ECLAT.

Otro algoritmo que permite generar los itemsets frecuentes es ECLAT. La diferencia principal entre ECLAT y Apriori es la estructura de los conjuntos de datos con los que trabaja cada uno. El algoritmo Apriori usa datos en formato horizontal, como se presentan el cuadro 2.1 mientras que el ECLAT también permite trabajar con datos en formato vertical, es decir, para cada item, se muestra en que transacciones se encuentra, ver cuadro 2.2.

item	TID
Pan	1,2,4,5
Leche	1,3,4,5
Huevos	1,5
Cerveza	2,3,4,5
Pañales	2,3,4

Cuadro 2.2: Transacciones en formato vertical.

Actualmente se están proponiendo diversos algoritmos, similares a ECLAT, que trabajan con datos verticalmente, ya que resultan más efectivos computacionalmente, que los enfoques horizontales. Fundamentalmente, la generación de itemsets sigue la misma mecánica que Apriori, pero trabajando con otra estructura de datos, que resulta mucho más rápida porque para la busqueda de itemsets frecuentes solo escanea la base de datos una vez mientras que Apriori lo hace varias veces. Esta ventaja del ECLAT frente Apriori, resulta evidente únicamente en conjuntos de datos no muy grandes, ya que en el caso de grandes conjuntos de datos, las listas verticales ocupan demasiada memoria y aparece el problema de la la escalabilidad del algoritmo, en estos casos resulta mejor usar Apriori.

2.1.6. Generación de las reglas de asociación

Una vez encontrados los k-itemsets frecuentes, se procede a obtener las reglas de asociación ocultas. Cada k-itemset, Y, puede generar $2^k - 2$ reglas de asociación, ignorando $Y \longrightarrow \emptyset$ y $\emptyset \longrightarrow Y$. La regla de asociación se genera efectuando una partición sobre Y en X e Y-X, siendo X no vacío, de forma que la regla de asociación $X \longrightarrow Y - X$ satisface el umbral de confianza *MinConf*. Veamos un ejemplo.

Ejemplo: Sea $X = \{1,2,3\}$ un itemset frecuente. Hay seis reglas de asociación candidatas, que

pueden generarse con un X no vacío: $\{1,2\} \longrightarrow \{3\}$, $\{1,3\} \longrightarrow \{2\}$, $\{2,3\} \longrightarrow \{1\}$, $\{1\} \longrightarrow \{2,3\}$, $\{2\} \longrightarrow \{3,1\}$ y $\{3\} \longrightarrow \{1,2\}$ Puesto que todos los itemsets satisfacen el umbral del soporte, las reglas de asociación también lo satisfacen. Para obtener la confianza de cada regla de asociación no es necesario escanear de nuevo el conjunto de transacciones, ya que disponemos de los soportes. Por ejemplo, si consideramos $\{1,2\} \longrightarrow \{3\}$, que se genera a partir del itemset frecuente $X = \{1,2,3\}$ la confianza de esta regla viene dada por $Supp(\{1,2,3\})/Supp(\{1,2\})$. La propiedad de anti-monotonía para el soporte asegura que ambos itemsets son frecuentes, por ello no es necesaria la lectura del conjunto de datos completo otra vez.

Sin embargo, la confianza, no verifica la propiedad de antimonotonía. Es decir, la confianza de $X \longrightarrow Y$ puede ser mayor o menor que la confianza de otra regla $\tilde{X} \longrightarrow \tilde{Y}$, donde $\tilde{X} \subseteq X$ y $\tilde{Y} \subseteq Y$. Sin embargo, si comparamos reglas generadas del mismo itemset Y, se tiene el siguiente resultado para la confianza:

Teorema 2. *Si una regla $X \longrightarrow Y - X$ no satisface el umbral de confianza, entonces ninguna regla $X' \longrightarrow Y - X'$ donde X' es un subconjunto de X, satisface el umbral de confianza tampoco.*

Esta propiedad, cuya demostración se puede consultar en [17] (pag. 350), permite generar reglas de forma eficiente, efectuando la poda de todas las reglas que se obtienen a partir de una regla cuya confianza no supera el $MinConf$.

2.1.7. Generación de reglas con el algoritmo Apriori

El algoritmo Apriori utiliza un enfoque por niveles, donde cada nivel se corresponde con el número de items que genera la consecuencia de cada regla. Inicialmente se extraen las reglas con alta confianza, es decir, las que tienen solo un item en la parte consecuente. Estas reglas se usan posteriormente para generar nuevas reglas candidatas.

Ejemplo: Sean $\{acd\} \longrightarrow \{b\}$ y $\{abd\} \longrightarrow \{c\}$ reglas de alta confianza, la regla candidata $\{ad\} \longrightarrow \{bc\}$ se genera uniendo las consecuencias de ambas reglas.

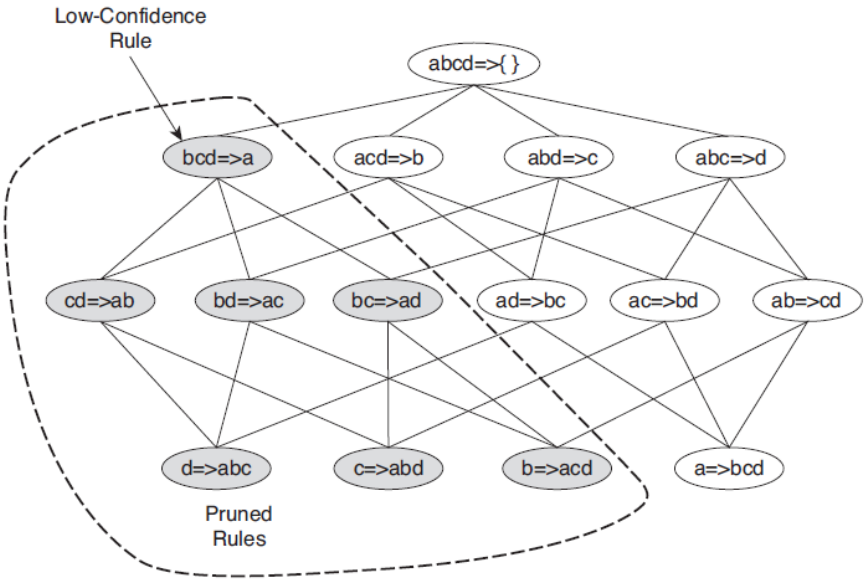


Figura 2.4: Generación de reglas con algoritmo Apriori

La figura 2.4 muestra las reglas de asociación generadas a partir del itemset $\{a,b,c,d\}$, si algún nodo tiene una confianza baja, entonces el subgrafo de reglas que se suceden tras esta, incluída ella misma, se podan como consecuencia del teorema anterior. Suponer que la confianza de $\{bcd\} \longrightarrow \{a\}$ es baja. en consecuencia todas las reglas que contengan en su consecuencia el item a tienen baja confianza y pueden ser eliminadas, como se ve en la figura 2.4.

El pseudocódigo para la generación de reglas se muestra en el algoritmo 4. Se puede observar una similitud entre el algoritmo 3 de generación de itemsets frecuentes y el que se presenta en algoritmo 4 con *ap-genrules*. La única diferencia es que en la generación de reglas no tenemos que realizar nuevos escaneos sobre la base de datos para computar la confianza de las reglas candidatas. En su lugar, se determina la confianza para cada regla usando el soporte contabilizado durante la generación de los itemsets frecuentes.

Algorithm 4 Generación de reglas con el algoritmo Apriori.

for *cada k-itemset frecuente* $F_k, k \geq 2$ **do**
  $H_1 = \{i | i \in f_k\}$ (1-items consecuentes de la regla)
  Llamada a $ap-genrules(f_k, H_1)$
end

Algorithm 5 Procedimiento $ap-genrules(f_k, H_m)$.

$k = |f_k|$ (tamaño del itemset frecuente)
$m = |H_m|$ (tamaño de la consecuencia de la regla)
if $k > m+1$ **then**
  $H_{m+1} = apriori-gen(H_m)$
  for *cada* $h_{m+1} \in H_{m+1}$ **do**
    $conf = \sigma(f_k)/\sigma(f_k - h_{m+1})$
    if $conf \geq minconf$ **then**
      output regla $(f_k - h_{m+1}) \longrightarrow h_{m+1}$
      else
        eliminar h_{m+1} de H_{m+1}
      end
    end
  end
  Llamada a $ap-genrules(f_k, H_{m+1})$
end

2.2. Fundamentación teórica de las reglas de asociación ponderadas (WARM)

Las reglas de asociación ponderadas surgen en situaciones en las que no tiene sentido asignar la misma importancia a todos los items de la cesta de la compra. Es decir, dependiendo de la situación, se deben elegir los pesos de forma flexible para resolver diferentes situaciones.

Ejemplo: en el contexto del mercado, algunos artículos como joyería o prendas de diseño, tienen mucho más valor económico, que otros como gominolas o chicles. Con los métodos no ponderados, vistos hasta ahora, las reglas que involucran joyas o artículos de lujo pueden tener un soporte más bajo que aquellas relacionadas con los dulces, y sin embargo son mucho más significativas a nivel económico que las otras, pues una sola venta puede suponer muchísimos más beneficios. En este caso, la ponderación a cada item no se asigna en función de la frecuencia con que aparecen sino a partir del beneficio por la venta de una unidad.

La primera formulación de un modelo basado en pesos la propone Ramkumar et al, [10] en 1997. Algoritmos como MINWAL (C.H.Cai, 1998) o WARM (F.Tao, 2003)[8], asumen que los pesos de cada item se conocen antes de obtener las reglas de asociación, lo cuál, es mucho suponer en situaciones reales.

El eficiente algoritmo HITS [7], no asume conocer los pesos sino que los extrae del conjunto de datos, para ello se introduce una nueva medida para el soporte que parte de la idea de que la importancia de un item es independiente de su frecuencia.

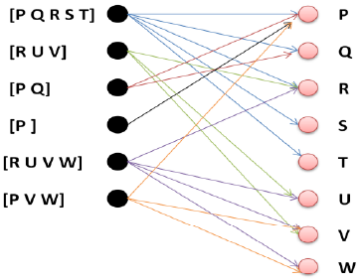
2.2.1. Algoritmo HITS para reglas de asociación ponderadas

El algoritmo HITS, introducido originariamente por Kleinberg, para determinar el page rank en Internet, fue adaptado al cálculo de pesos para un conjunto de datos, por Sun en 2008, [6],[7]. Se introduce el concepto de w-soporte, cuya idea fundamental es que un itemset frecuente puede no tener tanta importancia como parezca, dado que los pesos de las transacciones son diferentes de los pesos de los items que las componen. Estos pesos de las transacciones se obtienen directamente de la estructura interna de la base de datos.

Sea $D = \{T(1), T(2), ..., T(m)\}$ un conjunto de transacciones, $I = \{i(1), i(2), ..., i(n)\}$ sus items correspondientes. La base de datos asociada se puede representar como un grafo bipartito $G = (D, I, E)$ donde $E = \{(T, I) : i \in T, T \in D, i \in I\}$ como se muestra en la figura 2.5.

TID	Transacción
1	P,Q,R,S,T
2	R,U,V
3	P,Q
4	P
5	R,U,V,W
6	P,V,W

(a) Transacciones



(b) Grafo bipartito

Figura 2.5: Grafo bipartito de una base de datos.

Se puede observar que el soporte de un item es proporcional al grado del vértice correspondiente, lo que muestra que el soporte visto hasta ahora no considera la diferencia entre transacciones. Sin embargo, resulta fundamental que transacciones distintas tengan diferentes pesos. Una transacción buena, tiene una alta ponderación, debe contener buenos items, con alta ponderación, y análogamente un buen item debería estar en transacciones buenas. Por ello se puede establecer una analogía entre la relación existente entre las transacciones e items y los hubs y autoridades del algoritmo HITS.

En cada iteración, se calculan las autoridades y los hubs:

$$Auth(i) = \sum_{i \in T} Hub(T)$$

$$Hub(T) = \sum_{i \in T} Auth(i)$$

Cuando el algoritmo HITS converge se obtienen los pesos de hub de cada transaccion, que representan el potencial de cada transacción de contener items con buena valoración. Por ejemplo, una transaccion con pocos items puede tener buen hub si todos los items tienen muy buena valoración. En cambio, una

transacción con muchos items mal valorados puede tener un hub bajo.

Las reglas WARM se formulan en términos del w-soporte, que permite trabajar con hubs y autoridades. Se define el w-soporte para una transacción X como:

$$Wsupp(X) = \frac{\sum_{X \in T \wedge T \in D} hub(T)}{\sum_{T \in D} hub(T)}$$

El w-soporte de la regla de asociación $X \longrightarrow Y$ viene dado por wsupp($X \longrightarrow Y$)=wsupp($X \cup Y$).

Un itemset o transacción se dice que es significativa (buena) si su w-soporte es mayor que un umbral prefijado. En el caso de una regla de asociación, se mide si la aparición de X e Y conjuntamente resulta significativa, es decir w-supp para la regla supera un umbral.

La w-confianza de una regla se define como

$$wconf(X \longrightarrow Y) = \frac{wsupp(X \cup Y)}{wsupp(X)}$$

mide como de fuerte es la regla de asociación. Si $wconf(X \longrightarrow Y)$ tiene un valor alto, indica que hubs buenos que votan al itemset X también votan al Y, aunque la fracción de estos hubs resulte un valor pequeño.

En consecuencia, las reglas de asociación consisten en obtener todas las reglas cuyos w-soportes y w-confianzas superen un umbral prefijado en ambas.

El problema de obtener reglas de asociación ponderadas puede descomponerse, en dos fases, siguiendo un esquema muy similar al caso de reglas no ponderadas:

1. Encontrar todas las transacciones e itemsets significativos, cuyo w-supp supere el umbral *minw-Supp*.
2. Inducir reglas de asociación con los itemsets o transacciones obtenidos en el paso 1.

En el algoritmo 6, se presenta el pseudocódigo relativo a la primera etapa. Respecto a la inducción de reglas de asociación que se realiza en la segunda fase, no se adjunta el algoritmo, ya que no existe diferencia con lo presentado en el apartado correspondiente a reglas de asociación no ponderadas, salvo por la definición de la $wconf$.

2.3. Paquetes en R: arules

Una de las librerías más potentes con implementación de reglas de asociación es arules propuesta en 2005 por Hahsler, Grün y Hornik, ver [4]. Todas las funcionalidades del paquete se pueden consultar en [21].

Algunas de las funciones más utilizadas que permiten la obtención de reglas de asociación y que posteriormente se utilizarán son:

- **hits:** permite generar los pesos de una colección de transacciones usando el algoritmo HITS, que originariamente surgió para valorar la importancia de páginas web.
- **eclat:** permite determinar los itemsets frecuentes en conjuntos de datos dispuestos en formato vertical. La versión ponderada de ECLAT esta implementada mediante la función **weclat**, permite efectuar reglas de asociación ponderadas (WARM). Es necesario pasarle el vector de pesos asociados a los items, normalmente obtenido con la función hits.

Algorithm 6 Algoritmo HITS. Búsqueda de transacciones significativos.

Inicializar auth(i) desde 1 para cada item i
for *(l=0 hasta nº iteraciones)* **do**
  auth'(i)=0 para cada item i
  for *todas las transacciones* $t \in D$ **do**
    hub(t)=$\sum_{i \in t} auth(i)$
    auth'(i)+=hub(t) para cada item $i \in t$
  end
  auth(i)=auth'(i) para cada item i, normalizar auth
end
$L_1 = \{i | wsupp(i) \geq minwsupp\}$
for $(k = 2;\ L_{k-1} = \emptyset; k++)$ **do**
  $C_k = apriori - gen(L_{k-1})$
  for *todas las transacciones* $t \in D$ **do**
    $C_t = subset(C_k, t)$
    for *todos los candidatos* $c \in C_t$ **do**
      $c.wsupp+ = hub(t)$
    end
    $H+ = hub(t)$
  end
  $L_k = \{c \in C_k | c.wsupp/H \geq minwsupp\}$
end
Salida $= \cup_k L_k$

- **ruleInduction:** permite generar reglas de asociación a partir de un conjunto de itemsets obtenidos de las transacciones. La función ruleInduction permite generar reglas de asociación cerradas definidas en [8], como $X \longrightarrow Y$ donde X e Y son itemsets frecuentes cerrados, es decir, aquellos en los que ninguno de los conjuntos que los contienen tienen el mismo soporte que ellos y además dicho soporte supera el umbral. El uso de itemsets frecuentes y cerrados evita la redundancia de reglas de asociación logrando un algoritmo más eficiente.

  En la función ruleInduction existen diferentes métodos de generación de reglas implementados, que se pueden especificar, aunque el método por defecto es "ptree":

  - **apriori:** usa previamente la función apriori implementada para obtener el umbral de soporte y todas las reglas de asociación con itemsets de un elemento en la consecuencia de la regla. Posteriormente en un segundo paso elimina todas las reglas que no se generan con alguno de los itemsets. Este procedimiento resulta muy lento en el caso de itemsets con muchos elementos o con un soporte muy bajo.
  - **ptree:** está basado en algoritmos del tipo FP-growth. Como se menciona en [8] CLOSET es un caso particular de algoritmo FP-growth. Estos algoritmos tienen un enfoque radicalmente distinto al paradigma "generate and test" en el que se basan los algoritmos Apriori, que los hacen ser mucho más eficientes. En su lugar, construye una estructura de datos compacta, el FP-tree y extrae los itemsets frecuentes de esta estructura, para posteriormente generar reglas y calcular sus confianzas. Este método se puede aplicar sin las transacciones previamente calculadas (si existe un conjunto completo de itemsets frecuentes), lo cual hace el algoritmo más rápido, debido a que la estructura de ptree almacena los soportes de los itemsets y los recupera rápidamente para obtener las reglas. Si se aplica ptree aportando una matriz

de transacciones es más lento pero es necesario en caso de que el conjunto de itemsets frecuentes no sea completo.

En el caso práctico que se verá posteriormente sobre la Cohorte EpiChron, se obtienen reglas de asociación ponderadas. La razón de ponderarlas es que enfermedades con una prevalencia determinada afectan de distinta forma dependiendo de con qué otras patologías aparezcan, [18],[15]. Para ello, se asignan los pesos correspondientes a las transacciones, constituidas por "listados de enfermedades" asociados a cada paciente, mediante la función HITS. Se obtienen los itemsets o transacciones frecuentes, aplicando el algoritmo ECLAT ponderado, dado que el formato vertical de los datos así lo requiere. Por último, a partir de ruleInduction se generan las reglas frecuentes estableciendo unos criterios de confianza y soporte, para efectuar una poda posterior de las reglas redundantes.

Capítulo 3

Aplicación en estudios epidemiológicos longitudinales.

3.1. Descripción de la base de datos: Cohorte EpiChron.

La Cohorte EpiChron, para investigación en enfermedades crónicas [9], es la base de datos sobre la que se implementan las técnicas descritas en los capítulos anteriores. Esta cohorte ha sido conformada por el Grupo EpiChron de Investigación en Enfermedades Crónicas, perteneciente al Instituto de Investigación Sanitaria Aragón (IIS Aragón).

Dicha base de datos contiene información anonimizada sobre la población, usuaria del sistema de salud público de Aragón, durante los años 2010-2020, comprendiendo, en torno a 1.2 millones de personas, de las que consta la siguiente información a nivel de paciente:

- sociodemográfica: fecha de nacimiento, género, nacionalidad, país de nacimiento y zona básica de salud (rural o urbana).
- clínico-farmacológica: resultados en salud, como diagnósticos, tratamientos, pruebas diagnósticas y en su caso, fecha de defunción. También tratamientos farmacológicos, incluyendo las preinscripciones de medicamentos y su posología.
- uso de servicios de salud: visitas médicas a atención primaria, ingresos hospitalarios, ingresos en las Unidades de Cuidados Intensivos (UCI) y procedimientos diagnósticos y quirúrgicos.

La población objeto del presente estudio, es la denominada “ancianos jóvenes", con edades comprendidas entre los 65-75 años, sobre la que el objetivo médico final es la anticipación y prevención de aparición de enfermedades y de mala evolución de la salud.

Para ello, se analizará la información demográfica y clínica de esta subpoblación durante el periodo 2010-2019, excluyendo 2020, por ser un año no representativo, debido a las anormalidades y retrasos que provocó la pandemia del COVID-19.

Esta información esta estructurada en 10 conjuntos de datos, uno por año, entre 2010 y 2019. Contienen las siguientes variables: identificador anónimo de usuario, sexo, fecha de nacimiento, código asociado a la enfermedad, [22] ,fecha de primer registro en la historia clínica de dicha enfermedad, fecha de fallecimiento, fecha de baja en el sistema, años en el momento de fallecer y a partir de 2011 en adelante se añade la variable edad en el año de estudio. Observar que en los sucesivos conjuntos de datos a partir de 2011 incluido, únicamente aparecen las nuevas incorporaciones en el sistema, es decir, pacientes que durante ese año registraron una nueva enfermedad.

Cabe mencionar, como crítica hacia los futuros resultados que se obtendrán que, según los datos presentes en los ficheros aportados, existen comorbilidades anteriores a 2010, en las que no se especifica la fecha de diagnóstico exacta y su identificación suele coincidir con la fecha de nacimiento del paciente.

En la siguiente tabla se muestra el número de observaciones iniciales por fichero.

Fichero	Observaciones
2010	394139
2011	57845
2012	46683
2013	46536
2014	43148
2015	42102
2016	41518
2017	39734
2018	38097
2019	37780

3.2. Aplicación de técnicas a un estudio longitudinal sobre la Cohorte EpiChron.

Como se ha mencionado inicialmente, nuestro objetivo en relación al trabajo con la base de datos de la Cohorte EpiChron y más en particular sobre el segmento de la población de 65-75 años, es determinar trayectorias de multimorbilidad entre enfermedades que aparecen en patrones de multimorbilidad frecuentes, es decir, la secuencia de aparición de enfermedades, que podría darse a lo largo del tiempo. Estas trayectorias son modelizables a partir de los datos existentes en el periodo 2010-2019. Así mismo, se determinarán diferentes clusters de enfermedades.

Esquemáticamente se seguirá el siguiente proceso:

1. Procesado de los datos: Reagrupación y eliminación de enfermedades con baja prevalencia.
2. Determinación de clusters de enfermedades.
3. Reglas de asociación ponderadas entre enfermedades.
4. Generación de trayectorias de multimorbilidad.

3.2.1. Procesado de los datos: Reagrupación y eliminación de enfermedades con baja prevalencia.

El grupo EpiChron propone reagrupar enfermedades, ver cuadro 3.1. Partiendo del fichero de datos de 2010, tenemos inicialmente 155 enfermedades distintas, debido a la similitud entre diferentes variantes de una familia de enfermedades y cierto grado de discrecionalidad en la codificación por parte del facultativo, se efectúa la reagrupación de algunas, bajo los nombres de neoplasias, diabetes e hipertensión, de esta forma se consigue simplificar resultados, quedando un total de 89 códigos de enfermedades distintas. Esta reagrupación se efectuará en los ficheros de los 10 años.

Una vez efectuada la reagrupación, se estudia la prevalencia de cada enfermedad, sobre el fichero del año 2010, tal como indica EpiChron. La prevalencia para cada código de enfermedad se calcula como

$$prevalencia/enfermedad = \frac{pacientes \quad que \quad padecen \quad enfermedad}{total \quad de \quad pacientes \quad en \quad 2010} \cdot 100$$

Diabetes	49,50
Hipertensión	98,99
Neoplasias	11-47

Cuadro 3.1: Agrupación de códigos de enfermedades.

Una vez calculadas las prevalencias, tras valorar diversos umbrales mínimos de prevalencia, se ve aconsejable eliminar, aquellas enfermedades cuya prevalencia es inferior al 1 %. De esta forma aportamos mayor consistencia a los resultados obtenidos en los próximos pasos. Finalmente, el número de enfermedades con el que trabajaremos es 25. Se eliminan de los ficheros las observaciones de pacientes cuya patología no figura dentro de las 25 anteriores.

Por último para trabajar con todos los datos unimos los 10 ficheros en uno solo. En este fichero detectamos los pacientes que presentan una sola enfermedad a lo largo de los 10 años, ya que estos casos no son objeto de estudio, pues no son multimórbidos. Tras realizar un depurado de estos pacientes, con una sola enfermedad, obtenemos un fichero con 597213 registros de enfermedades de un total de 122248 pacientes con multimorbilidad, de los cuales 56708 son hombres y 65540 son mujeres. Sobre este fichero se obtendrán clusters y se calculan reglas de asociación ponderadas.

3.2.2. Determinación de clusters de enfermedades.

Las 25 enfermedades de nuestro fichero depurado se pueden agrupar en diferentes clusters de multimorbilidad, se corresponden a agrupaciones de enfermedades con características similares. En este caso al ser fuzzy clustering se le asignara a cada enfermedad una probabilidad de pertenencia a cada cluster.

Para realizar los clusters de enfermedades, se deben preparar los datos. Para ello usando la información del fichero depurado que contiene los registros de todas las enfermedades, se construirá una matriz binaria en cuyas filas figuran 122248 pacientes distintos y en sus columnas hay 24 enfermedades para cada sexo, debido a que hay enfermedades específicas de cada sexo, de forma que, en el caso femenino no existe problemas de próstata, mientras que en el masculino no hay desordenes menopáusicos.

Previamente a la aplicación de los métodos fuzzy, se estudia cuál podría ser el número óptimo de clusters. Para ello se usa la librería ClustOfVar, que permite realizar hard-clustering, sobre variables, aplicando métodos jerárquicos y de k-medias. Resulta interesante para nuestro análisis porque permite el uso de variables cuantitativas, cualitativas o mixtas.

En particular, se hace uso de la función hclustvar, que realiza un cluster jerárquico. En este caso, dentro de la distinción de variables en los argumentos de la función, se indica que todas son cualitativas, pues al ser binarias las consideramos de tipo factor. Las figuras 3.1 y 3.2 representan el dendograma, resultado por sexos del cluster jerárquico.

La función stability, evalúa la estabilidad de las particiones jerárquicas realizadas mediante técnicas de Bootstrap. En este caso, nos permite determinar, los valores de número de clusters para los que se maximiza el índice Rand, que permite comparar la similitud de resultados para diferentes particiones. Fijaremos la semilla para reproducibilidad de resultados y efectuaremos 10 remuestreos, usando la totalidad de pacientes en cada sexo. En las figuras 3.3, 3.4 se observa que el número óptimo de clusters podría ser tanto en el caso masculino como en el femenino 4.

Efectuamos una partición en el número de clusters aconsejado para cada sexo, mediante la función cutree. En la salida adjunta en los cuadros 3.2 y 3.3 se observa el coeficiente “squared loading", que in-

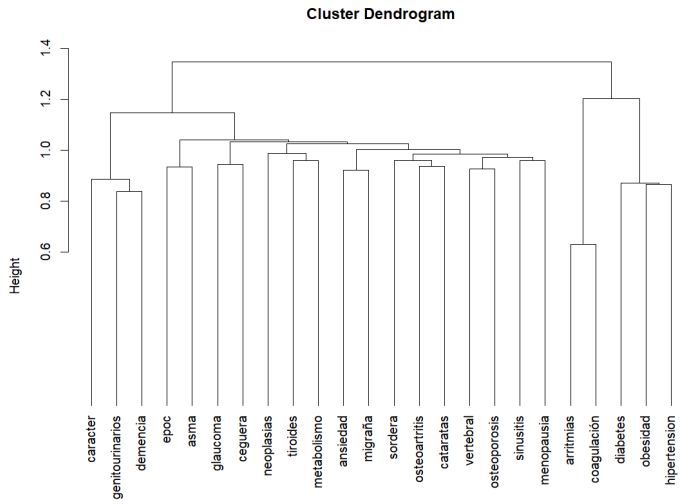


Figura 3.1: Dendograma o cluster jerárquico para las 24 enfermedades en sexo femenino.

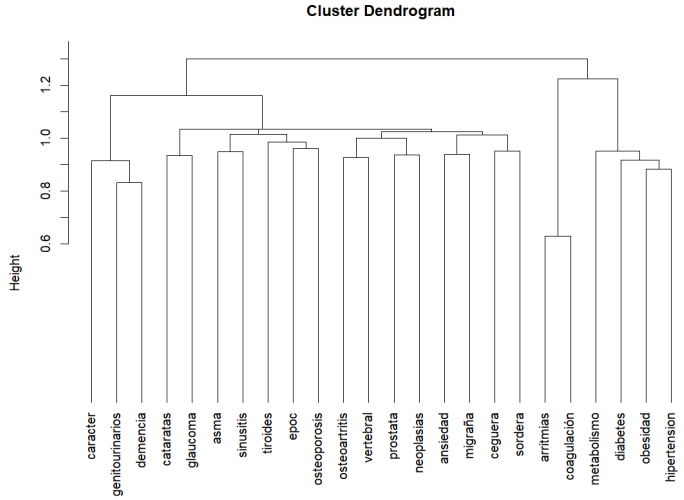


Figura 3.2: Dendograma o cluster jerárquico para las 24 enfermedades en sexo masculino.

dica el porcentaje de variabilidad explicada por las dos primeras componentes asociadas al caso binario (análisis de correspondencias múltiples) creadas por el método de clusterización que hemos elegido en este caso. Se puede observar que hay casos, por ejemplo, neoplasias tanto en hombres como mujeres o trastornos metabólico-lipídicos en mujeres, en los que la enfermedad apenas viene representada debido a que es muy bajo el valor del “squared loading” y en consecuencia no esta clara su pertenencia a un clúster concreto. Por ello resulta natural, efectuar un cluster fuzzy de caras a averiguar las probabilidades de pertenencia a cada cluster, especialmente de estas enfermedades con baja representación.

Para realizar el cluster fuzzy aplicaremos la función NEFRC, de la librería fClust, adecuada para datos relacionales, es decir, es necesario aportar su matriz de distancias. Entre los argumentos requeridos esta la matriz de distancias mencionada, que debe construir con la métrica más adecuada, [3]. Entre otras métricas se ha probado a usar la distancia de “jaccard”, “simpson” o “Kulczynski2”, todas

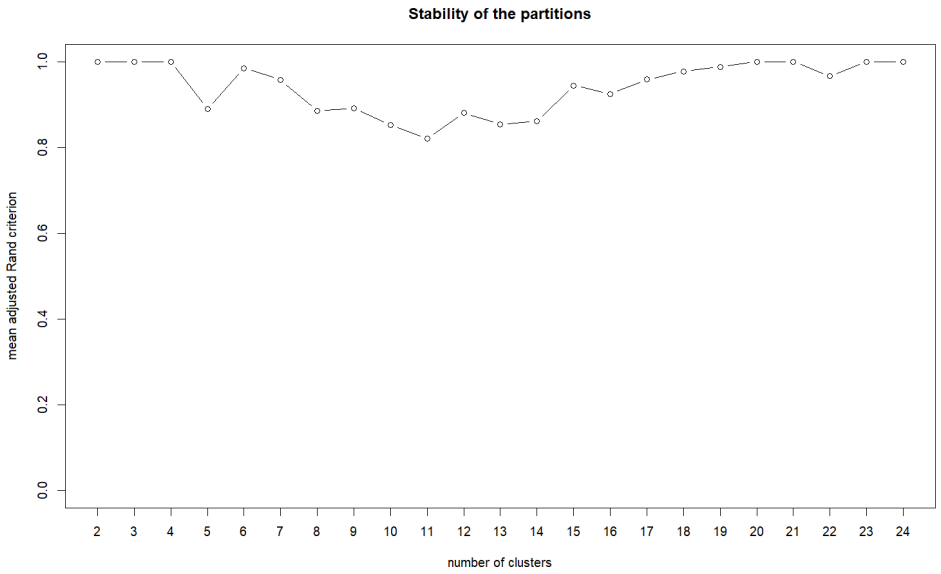


Figura 3.3: Estudio del número óptimo de clusters para el caso de mujeres, se elige 4 clusters.

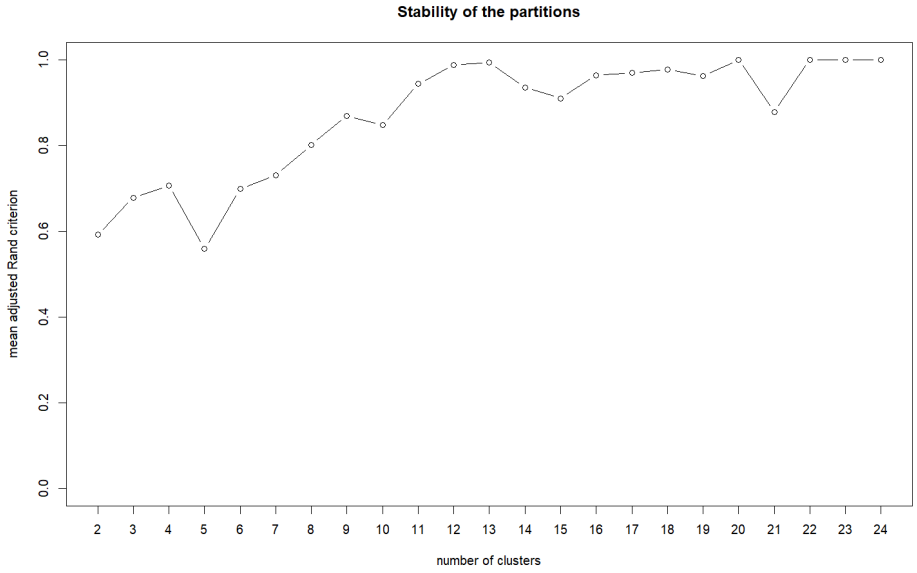


Figura 3.4: Estudio del número óptimo de clusters para el caso de hombres, se elige 4 clusters.

adecuadas para datos binarios. Se han calculado las matrices de distancias asociadas y se ha estudiado que existiera correlación entre sus elementos, además se ha buscado que tuvieran cierta variabilidad de caras a aportar mayor riqueza en la determinación de las pertenencias.

Tras realizar varias pruebas, la distancia elegida es la de “Kulczynski2", calculada como el promedio de las probabilidades condicionadas de que las enfermedades (dos a dos) estén presentes. Del mismo modo, se han realizado diversas pruebas con el parámetro de difusión m y se ha observado que para valores por encima de 1,2 todas las variables se quedan sin asignar, en particular, para $m = 1{,}14$ se obtiene la clasificación más acorde al caso hard-clustering.

Tras aplicar la función NEFRC sobre la matriz de distancias en cada sexo, para cuatro clusters, parámetro de difusión 1,14, 10 reinicios aleatorios e índice de validación fuzzy silueta, se obtienen

Clusters	Squared Loading
Mujeres	
Cluster 1	
arritmias	0.68
coagulación	0.68
Cluster 2	
problemas vertebrales	0.173
osteoartritis	0.149
osteoporosis	0.146
migraña	0.144
cataratas	0.131
ansiedad	0.122
desórdenes menopausia	0.109
sinusitis, catarro	0.087
sordera	0.086
asma	0.074
metabolismo-lípidicos	0.051
ceguera	0.049
tiroides	0.041
epoc(pulmonar)	0.016
glaucoma	0.014
neoplasias	0.005
Cluster 3	
demencia	0.45
genitourinarios	0.45
desórdenes de carácter	0.37
Cluster 4	
obesidad	0.43
hipertensión	0.42
diabetes	0.41

Cuadro 3.2: Resultados agrupación hard-clustering para mujeres y squared loading asociado a cada enfermedad del cluster.

los resultados que se muestran en los cuadros 3.5, 3.4, donde se observa la partición más probable en función de las matrices de pertenencia (véase los cuadros 3.6, 3.7).

Si establecemos una comparativa entre el cluster hard obtenido y el fuzzy se observa que las enfemedades que tienen un squared loading alto se mantienen en los grupos con una alta probabilidad, en cambio las de squared loading bajo tienen unas probabilidades de pertenencia repartidas entre dos o más clusters.

En el caso de mujeres podemos ver que el cluster 2 está claramente definido, formado por arritmias y coagulación. El cluster 4 esta compuesto por enfermedades genitourinarias, desórdenes cognitivos (demencia) y desórdenes de carácter o del estado de ánimo (depresión, bipolaridad) que aparecen con una probabilidad de pertenencia muy alta, mas del 85 %. En el cluster 1 se dan conjuntamente obesidad, hipertensión, diabetes y trastornos metabólico-lipídicos, glaucoma con probabilidades de más de más del 60 %. Si observamos el cuadro 3.6 el resto de enfermedades tiene probabilidades de pertenencia muy repartidas entre los clusters 1,3, 4, como es el caso de asma, neoplasias o EPOC (enfermedad pulmonar obstructiva crónica). También podemos observar, en el cluster 3 que aparecen en grupo las

Clusters	Squared Loading
Hombres	
Cluster 1	
arritmias	0.69
coagulación	0.69
Cluster 2	
problemas vertebrales	0.199
osteoartritis	0.157
próstata	0.139
cataratas	0.122
sordera	0.100
sinusitis	0.097
migrañas	0.093
ansiedad	0.100
asma	0.067
osteoporosis	0.055
ceguera	0.055
epoc (pulmonar)	0.039
neoplasias	0.033
tiroides	0.024
glaucoma	0.023
Cluster 3	
demencia	0.47
genitourinarios	0.47
desórdenes de carácter	0.32
Cluster 4	
obesidad	0.37
hipertensión	0.35
diabetes	0.32
metabolismo-lípidicos	0.21

Cuadro 3.3: Resultados agrupación hard-clustering para hombres y squared loading asociado a cada enfermedad del cluster.

enfermedades relacionadas con problemas óseos: osteoporosis y problemas vertebrales. Sin embargo, la osteoratritis tiene una probabilidad más alta de estar en el cluster 1, que en el cluster 3, que en cualquier caso inferior es al 50 % en ambos. El resto de patologías del cluster 3, tampoco tienen una pertenencia asignación muy clara, probablemente porque clínicamente no estén asociadas a ninguno de los grupos mejor perfilados, es decir, su causa de aparición está poco relacionada con el resto de patologías del clúster. Es el ejemplo de las cataratas, la ceguera, la sordera.

En el caso masculino, es prácticamente análogo, queda perfectamente definidos los clusters 1,3 y 4. Observar que en este caso las enfermedades respiratorias EPOC y asma, las ubica en el segundo clúster junto con sinusitis, aunque en el caso de EPOC sigue estando muy equilibrada la probabilidad entre los clúster 2 y 4. Los problemas de tiroides y osteoartritis pueden estar en cualquiera de los tanto en el clúster 2 como en el 4 primeros clusters, si bien puede ser más probable en el 4.

Mujeres
Cluster 1
asma, osteoartritis, obesidad, metabolismo-lípidos , glaucoma, diabetes, hipertensión, neoplasias
Cluster 2
arritmias, coagulación
Cluster 3
sinusitis, desórdenes menopausia, vertebrales, osteoporosis, tiroides, ansiedad, migraña, cataratas, ceguera, sordera
Cluster 4
epoc,genitourinarios, demencia y desórdenes de carácter

Cuadro 3.4: Resultados clasificación en clusters más probable en mujeres tras realizar fuzzy cluster con 4 grupos.

Hombres
Cluster 1
arritmias, coagulación
Cluster 2
epoc, asma, sinusitis, próstata, osteoporosis, vertebrales, ansiedad, migraña, cataratas, ceguera, sordera.
Cluster 3
genitourinarios, demencia, desórdenes de carácter
Cluster 4
osteoartritis, obesidad, tiroides, metabolismo-lípidos, glaucoma, diabetes, hipertensión, neoplasias

Cuadro 3.5: Resultados clasificación en clusters más probable en hombres tras realizar fuzzy cluster con 4 grupos.

3.2.3. Reglas de asociación ponderadas entre enfermedades.

Tras un primer estudio preliminar, con la asignación de clusters se generan reglas de asociación entre enfermedades. Para ello se hace uso de la librería arules. Además clínicamente conviene realizar una separación por sexos de caras a analizar los resultados [18], [19], [15].

A partir del fichero depurado que contiene los registros de todos los años se realiza una lectura de las transacciones donde cada individuo tendrá asignado el “listado” de enfermedades que padece entre 2010 y 2019. Como se puede observar en el cuadro 3.8.

Mediante el algoritmo HITS se asignan los pesos de ponderación (weight) a cada itemset, que se añaden a la información disponible para las transacciones.

Con el algoritmo ECLAT ponderado se generan los itemsets frecuentes. Las reglas de asociación se generan con ruleInduction y tras ello se realiza una poda de las reglas redundantes. Para ello se impone que la confianza sea del 50 %, el soporte del 1 % y la ordenación por lift.

Matriz de pertenencia para mujeres				
Enfermedad	Clus 1	Clus 2	Clus 3	Clus 4
asma	0.39	0.01	0.34	0.26
osteoartritis	0.47	0.00	0.33	0.20
obesidad	0.81	0.00	0.09	0.09
metabolismo-lipídicos	0.67	0.00	0.22	0.11
glaucoma	0.61	0.01	0.22	0.17
diabetes	0.81	0.00	0.08	0.11
hipertensión	0.92	0.00	0.04	0.03
neoplasias	0.35	0.01	0.33	0.31
arritmias	0.00	1.00	0.00	0.00
coagulación	0.00	1.00	0.00	0.00
sinusitis	0.20	0.01	0.63	0.16
menopausia	0.20	0.01	0.63	0.17
vertebral	0.25	0.00	0.51	0.23
osteoporosis	0.11	0.00	0.76	0.13
tiroides	0.38	0.01	0.38	0.23
ansiedad	0.22	0.00	0.59	0.19
migraña	0.20	0.01	0.56	0.23
cataratas	0.30	0.00	0.47	0.23
ceguera	0.23	0.01	0.51	0.25
sordera	0.27	0.01	0.43	0.29
epoc	0.30	0.01	0.30	0.38
genitourinarios	0.07	0.00	0.05	0.87
demencia	0.05	0.0	0.06	0.89
desórdenes de carácter	0.07	0.00	0.08	0.85

Cuadro 3.6: Matriz de pertenencias para cada cluster caso femenino.

Con estos parámetros la longitud de las reglas obtenidas oscilan entre 1 y 5, de esta forma se puede observar la secuencia de aparición de enfermedades. En el fichero de hombres se generan bajo esas condiciones 1940, tras la poda quedan 752 reglas. En el caso de mujeres, se generan 3998 reglas y tras la poda quedaron 1512.

En el cuadro 3.9 se puede observar un extracto de las primeras reglas obtenidas para cada sexo. Observar en el caso masculino, que la mayoría de enfermedades están muy relacionadas con patologías cardiológicas. En todos los casos, la confianza y el soporte superan los limites impuestos llegando incluso al 60 % de confianza. Respecto al lift, por ejemplo, observamos que en el caso de las tres primeras reglas aparece 4 veces de forma más frecuente que lo esperado, es decir, el soporte observado es 4 veces más frecuente que el que le correspondería supuesta independencia de los items, en más del 1 % de la población masculina estudiada. Observar además que se puede apreciar una secuenciación ya que padeciendo arritmias e insuficiencia cardiaca la próxima enfermedad que desencadena la coagulación es o diabetes o trastornos de metabolismo. Si en lugar de padecer insuficiencia cardiaca se detecta la enfermedad pulmonar EPOC, en el caso de arritmias y diabetes, también el consecuente es la coagulación. En otro sentido aparece la sexta y séptima regla, donde tras la gota y coagulación pueden aparecer arritmias y si aparece una tercera enfermedad que las provoque esta puede ser hipertensión.

A nivel de reglas para el sexo femenino, se puede apreciar mucha más similitud y conexión entre

Matriz de pertenencia para hombres				
Enfermedad	Clus 1	Clus 2	Clus 3	Clus 4
arritmias	1.00	0.00	0.00	0.00
coagulación	1.00	0.00	0.00	0.00
epoc	0.01	0.43	0.15	0.41
asma	0.01	0.58	0.12	0.29
sinusitis	0.01	0.63	0.11	0.25
próstata	0.01	0.62	0.11	0.26
vertebral	0.01	0.58	0.12	0.29
osteoporosis	0.01	0.53	0.17	0.28
ansiedad	0.01	0.51	0.19	0.28
migraña	0.01	0.55	0.16	0.28
cataratas	0.01	0.52	0.10	0.38
ceguera	0.01	0.60	0.13	0.26
sordera	0.01	0.57	0.13	0.29
genitourinarios	0.00	0.03	0.94	0.03
demencia	0.00	0.03	0.94	0.03
desórdenes de carácter	0.00	0.05	0.91	0.04
osteoartritis	0.01	0.45	0.10	0.45
obesidad	0.01	0.13	0.06	0.80
tiroides	0.02	0.37	0.17	0.44
metabolismo-lipídicos	0.00	0.15	0.04	0.81
diabetes	0.00	0.11	0.08	0.81
glaucoma	0.01	0.28	0.12	0.59
hipertensión	0.00	0.08	0.01	0.91
neoplasias	0.01	0.36	0.21	0.41

Cuadro 3.7: Matriz de pertenencias para cada cluster caso masculino.

items	transaction ID	weight
arritmias, obesidad, diabetes, hipertension	//+FpNQKATOR	1.709835e-05

Cuadro 3.8: Ejemplo de transacción

todas las primeras reglas, además de una confianza superior al 60 % en todos los casos. Puesto que el lift es mayor de 4 en todos los casos dichas reglas aparecen 4 veces más veces que en una situación de independencia entre los items, en más del 1 % de la población femenina estudiada. Según se observa, pequeños intercambios entre osteoartritis, coagulación, cataratas, hipertensión, genitourinarios, diabetes, obesidad y problemas de metabolismo se presentan de forma conjunta con arritmias.

Para posteriormente detectar la trayectoria de multimorbilidad más probable asociada al patrón proporcionado por una regla de asociación, se eligen las reglas que se muestran en 3.10, siguiendo los siguientes criterios, para cada sexo, con longitud máxima de 5 items, con el lift más alto, todas con confianza mayor del 50 %, que exista cierta relación entre ellas a la vez que se busca variabilidad entre las enfermedades que conforman dichas reglas con el objetivo de aportar mayor riqueza a los resultados.

Cabe mencionar que ha resultado más fácil encontrar reglas cumpliendo las condiciones en el caso femenino que en el caso masculino, ello se debe en parte al gran número de reglas de longitud menor de 5 en este último caso.

Primeras reglas de asociación obtenidas por sexos				
Antecedente	Consecuente	Soporte	Confianza	Lift
Hombres				
arritmias, insuficiencia cardiaca, diabetes	coagulación	0.011	0.604	3.989
arritmias, insuficiencia cardiaca, metabolismo	coagulación	0.012	0.604	3.987
arritmias, insuficiencia cardiaca	coagulación	0.021	0.583	3.852
arritmias, pulmonar, diabetes	coagulación	0.010	0.517	3.416
infarto miocardio, arritmias, metabolismo	coagulación	0.011	0.513	3.389
gota, coagulación, hipertensión	arritmias	0.014	0.648	3.084
gota, coagulación	arritmias	0.016	0.643	3.062
Mujeres				
genitourinarios, osteoartritis, coagulación, hipertensión	arritmias	0.010	0.619	4.288
osteoartritis, coagulación, cataratas, hipertensión	arritmias	0.011	0.619	4.287
osteoartritis, coagulación, diabetes	arritmias	0.010	0.612	4.238
genitourinarios, metabolismo, coagulación, hipertensión	arritmias	0.011	0.612	4.235
metabolismo, coagulación, diabetes, hipertensión	arritmias	0.012	0.606	4.198
osteoartritis, obesidad, coagulación, hipertensión	arritmias	0.010	0.605	4.192

Cuadro 3.9: Primeras reglas de asociación por sexos.

Reglas de asociación elegidas				
Antecedente	Consecuente	Soporte	Confianza	Lift
Hombres				
epoc, metabolismo, hipertensión,coagulación	arritmias	0.011	0.631	3.005
próstata,coagulación, diabetes,hipertensión	arritmias	0.010	0.629	2.991
genitourinarios, metabolismo, cataratas, hipertensión	diabetes	0.012	0.553	1.438
próstata, glaucoma, diabetes, hipertensión	metabolismo	0.011	0.711	1.141
obesidad, metabolismo, cataratas, hipertensión	diabetes	0.020	0.522	1.359
Mujeres				
genitourinarios, osteoartritis, coagulación, hipertensión	arritmias	0.010	0.619	4.288
osteoartritis, coagulación, cataratas, hipertensión	arritmias	0.011	0.619	4.287
genitourinarios, metabolismo, coagulación, hipertensión	arritmias	0.011	0.612	4.236
genitourinarios, metabolismo, coagulación, hipertensión	arritmias	0.010	0.606	4.192
obesidad, cognitivos, carácter, hipertensión	genitourinarios	0.011	0.527	1.996

Cuadro 3.10: Reglas de asociación elegidas.

Estos grupos de enfermedades aparecen en individuos de entre 65-75 años de forma frecuente. Por ello, las reglas anteriores nos permitirán, configuran patrones de multimorbilidad para generar una secuencia habitual de diagnóstico de enfermedades, teniendo en cuenta la fecha de detección.

3.2.4. Generación de trayectorias de multimorbilidad

Como se ha mencionado inicialmente nuestro objetivo es obtener la trayectoria de morbilidad, asociada a las enfermedades que conforman las reglas de asociación elegidas, es decir, determinar la secuencia de diagnóstico de enfermedades más probable para un individuo que padezca dos o más enfermedades de ese grupo.

Para ello se procede para cada grupo de enfermedades asociado a cada regla de la siguiente forma: a partir del fichero principal se filtran los registros de pacientes que han padecido dos 2 o más enfermedades de dicho grupo. A continuación, se construye un fichero en el que aparecen los pacientes en las filas y en las columnas las enfermedades y en cada celda se asigna un valor numérico que indica la posición por orden de aparición, información aportada por la fecha de detección de la enfermedad. La secuencia temporal de diagnóstico para un patrón, se establece obteniendo la media de los rangos de cada enfermedad.

Se desarrolla un método de estimación de la variabilidad de los rangos medios mediante intervalos de confianza bootstrap. En el remuestreo Bootstrap se generan 1000 réplicas o muestras y para cada una se calcula el estadístico, en nuestro caso el rango medio para cada código de enfermedad. Para estudiar su variabilidad se aportan los extremos del intervalo generado para dicho estadístico.

A continuación se procede a calcular los intervalos de confianza Bootstrap, para las reglas con mayor lift tanto en hombres como mujeres. Se puede observar en los cuadros 3.11 y 3.12 que los intervalos de confianza obtenidos para cada rango medio no se solapan, lo cuál indica que en todas las remuestras que se han efectuado en cada caso se sigue la misma secuencia de orden en la detección de las enfermedades, se mantienen los rangos medios establecidos para el patrón de comorbilidad. Se puede apreciar además que los intervalos resultan bastante simétricos respecto de la mediana.

Bootstrap para rangos medios en una regla para hombres					
Patrón	hipertensión	metabolismo	epoc	arritmias	coagulación
Rango medio	1.480	1.657	2.061	2.295	2.680
2.5 %	1.472	1.649	2.040	2.278	2.658
50 %	1.479	1.658	2.061	2.296	2.680
97.5 %	1.487	1.666	2.082	2.312	2.704

Cuadro 3.11: Intervalo de confianza Bootstrap para el rango medio en la regla elegida para hombres.

Bootstrap para rangos medios en una regla para mujeres					
Patrón	hipertensión	osteoartritis	arritmias	genitourinarios	coagulación
Rango medio	1.323	1.774	2.239	2.359	2.586
2.5 %	1.318	1.765	2.221	2.345	2.559
50 %	1.323	1.774	2.239	2.359	2.587
97.5 %	1.330	1.782	2.259	2.373	2.612

Cuadro 3.12: Intervalo de confianza Bootstrap para el rango medio en la regla elegida para mujeres.

Se obtiene la secuencia de detección para cada patrón de comorbilidad asociado a cada regla y se representa mediante un grafo dirigido como se muestra en la figura 3.5 y en la figura 3.6, en ambos casos el inicio de la secuencia es desde el nodo de hipertensión y la longitud de todos los caminos es de 5 nodos. Cabe remarcar que en los grafos sucesivos el nodo con la etiqueta “pulmonar”, se refiere a EPOC, el nodo con la etiqueta “cardiacos” se concreta en arritmias, “cognitivos” se refiere a desórdenes cognitivos o demencias y “carácter” a desordenes de carácter o del estado de ánimo que incluye depresiones o bipolaridad.

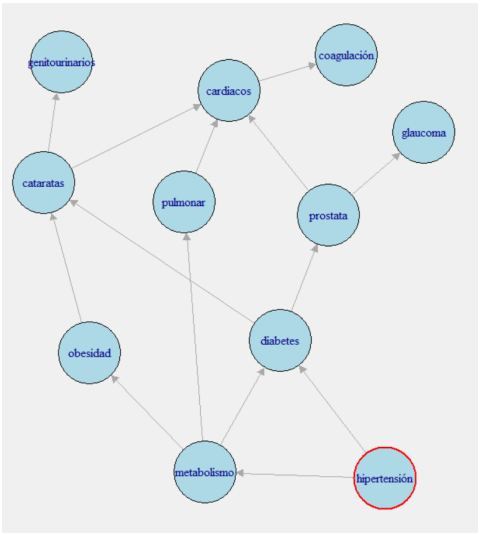


Figura 3.5: Trayectorias de multimorbilidad obtenidas para hombres usando rangos medios sobre las reglas de asociación elegidas.

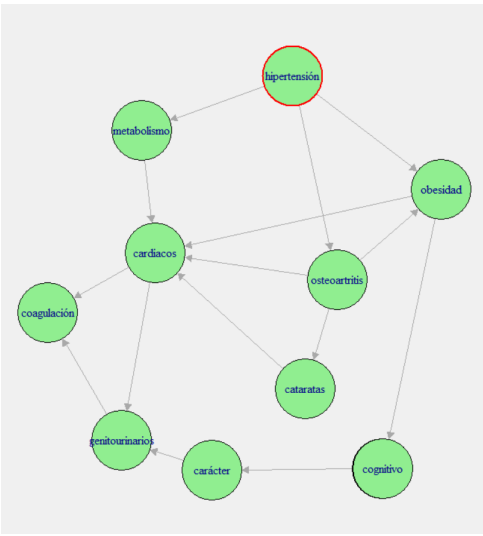


Figura 3.6: Trayectorias de multimorbilidad obtenidas para mujeres usando rangos medios sobre las reglas de asociación elegidas.

Grafos de flujo entre enfermedades de un patrón de comorbilidad

Se debe tener en cuenta que estas trayectorias obtenidas no son algo rígido e inamovible, como pueda dar la impresión al obtener rangos medios de cada enfermedad en cada patrón de comorbilidad. El rango medio indica que en la mayoría de personas se detecta la enfermedad en el orden obtenido, sin embargo, existen casos donde dicho orden cambia, es decir hay flujo de pacientes en otros sentidos.

Para ejemplificar lo anterior, se estudian los movimientos entre enfermedades de forma precisa, en el caso de la trayectoria con mayor lift, tanto para hombres como para mujeres.

Se procede trabajando con el primer patrón, el de mayor lift, para cada sexo. Filtramos el conjunto de datos con las enfermedades a estudio y obtenemos un fichero con su posición numérica en relación a la fecha de aparición. Se genera un grafo dirigido con pesos en los arcos de la siguiente forma:

- Los orígenes y destinos son las enfermedades del patrón, que constituyen los nodos.

- Los pesos de los arcos se calculan con un bucle en el que el peso (grosor) del arco se incrementa en una unidad, si existe un arco que va de una enfermedad dada a otra, siguiendo el orden secuencial existente en el fichero, para cada paciente.
- No se permiten los arcos recurrentes por no tener sentido clínico en este caso.

A partir de los pesos de la estructura de grafo anterior podemos obtener probabilidades absolutas y relativas:

- Absolutas: dividiendo el grosor del arco (número de individuos con ese arco) entre el grosor total de arcos.
- Relativas: calculando las probabilidades condicionadas al nodo origen, es decir, dividiendo el grosor del arco entre el grosor total de arcos para un origen concreto.

En la figura 3.7 se puede observar las probabilidades absolutas y relativas para el patrón de comorbilidad elegido (mayor lift) en mujeres, formado por hipertensión, osteoartritis, arritmias (cardiacos), alteraciones en la coagulación de la sangre y problemas genitourinarios. Se representan únicamente los arcos con probabilidad relativa superior al 14 % y mayor al 1,6 % en absolutas.

En el caso del grafo con probabilidades absolutas, se puede apreciar que un 27 % del total de mujeres dentro de este patrón se les diagnostica la osteartritis después de la hipertensión o que a un 13 % se les diagnostica la hipertensión después de la osteortritis. En cualquier caso, resulta evidente que más del 40 % de las pacientes fluctúa entre ambas enfermedades, en consecuencia están fuertemente conectadas.

En el caso de las probabilidades relativas para el mismo patrón, estaríamos hablando de una matriz de transición de probabilidades de primer orden para un grupo concreto. Se observa que la probabilidad de que la siguiente enfermedad diagnosticada de una mujer con hipertensión sea osteoartitis es de casi un 53 %, de que padezca arritmias es de un 15 % o de que se le diagnostiquen problemas genitourinarios un 26 %.

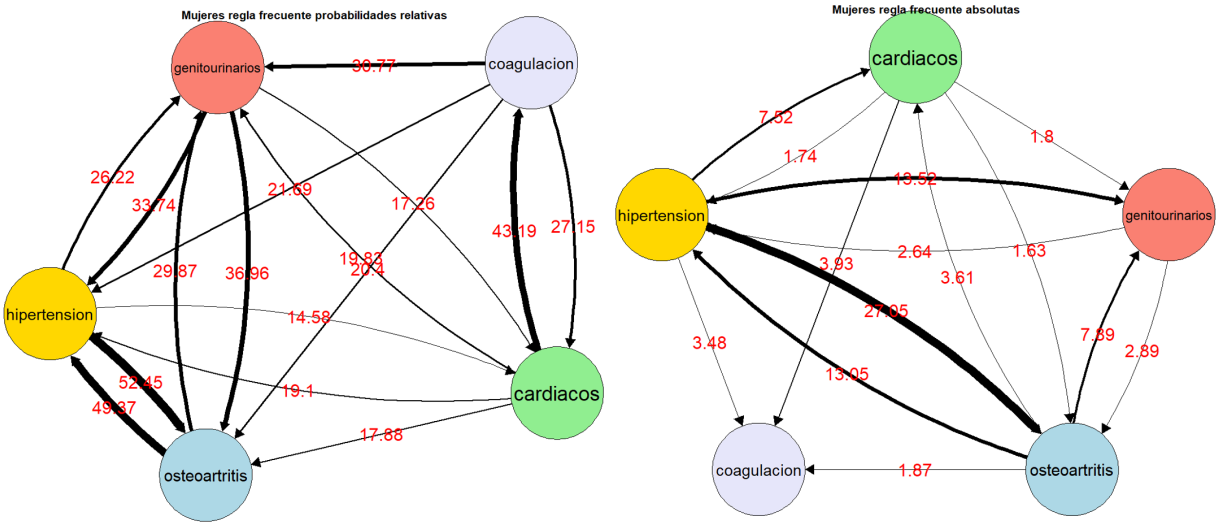


Figura 3.7: Grafos de probabilidades relativas y absolutas para el patrón elegido (mayor lift) en mujeres. Se representan únicamente los arcos con probabilidad relativa superior al 14 % y mayor al 1,6 % en absolutas.

En el caso, de los varones ver figura 3.8, el patrón elegido (el de mayor lift) consta de hipertensión, problemas del metabolismo-lipídico, arritmias, alteraciones en la coagulación y EPOC (pulmonar). Se

representan únicamente los arcos con una probabilidad mayor al 2 % en absolutas y mayor al 15 % en relativas.

En el caso de probabilidades absolutas, casi el 25 % del total de hombres con alguna enfermedad contenida en ese patrón se les diagnostica problemas de metabolismo después de hipertensión o casi un 20 % se les diagnostica la hipertensión después de problemas de metabolismo. Análogamente al caso femenino se observa que el 44 % de los pacientes varones fluctúa entre ambas enfermedades, en consecuencia están fuertemente conectadas.

Para el grafo con probabilidades relativas se observa que la probabilidad de que para una persona que padece arritmias (cardiacos), la siguiente enfermedad diagnosticada sea problemas en coagulación es del 53 % o de que padezca trastornos metabólicos del 19 %.

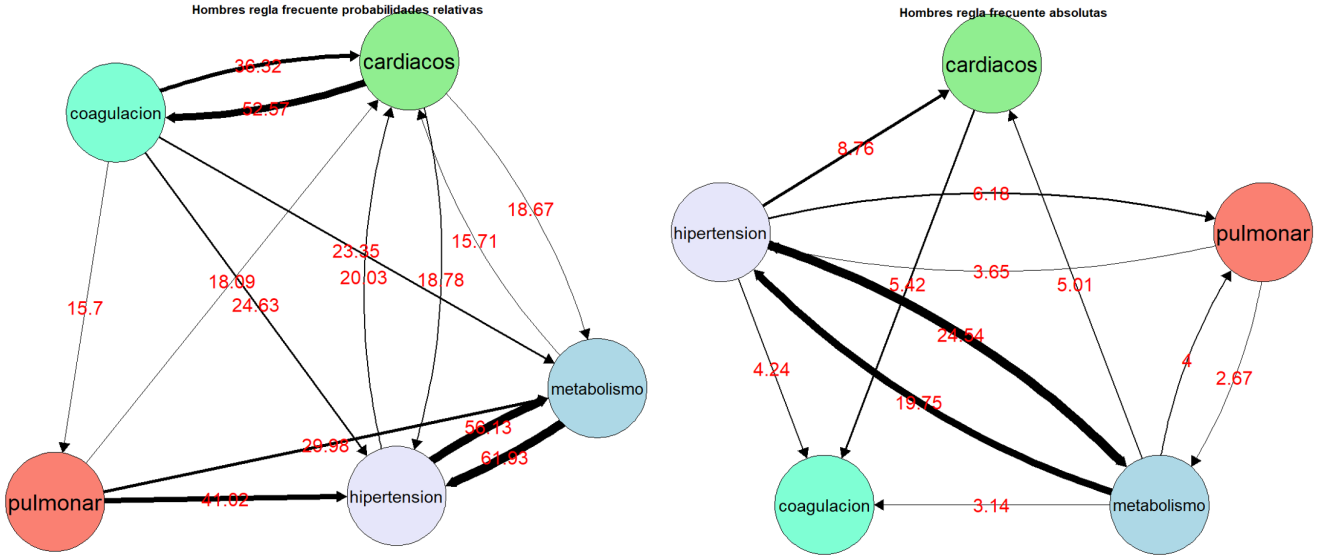


Figura 3.8: Grafos de probabilidades relativas y absolutas para el patrón elegido (mayor lift) en hombres. Se representan únicamente los arcos con una probabilidad mayor al 2 % en absolutas y mayor al 15 % en relativas.

Grafo clausura y matriz de transición de probabilidades de primer orden.

Como se ha visto para cada sexo las reglas/patrones escogidos están muy asociados entre si, por ello resulta natural estudiarlas todas de forma conjunta, de forma análoga a lo realizado en el apartado anterior con una regla concreta pero tomando todas las enfermedades que componen las reglas, lo llamaremos clausura.

El procedimiento de construcción de grafos se generaliza, con la única diferencia de que filtraremos nuestros registros de pacientes con dos o más enfermedades, considerando las de la clausura de todos los patrones para cada sexo, en el caso de los hombres serán 11 enfermedades y en el caso de las mujeres son 10.

A nivel de cifras, el total de pacientes distintos (no de registros) que contiene el fichero base es de 122248. La cantidad de pacientes hombres con dos o más enfermedades de la clausura es de 47273, que representan el 39 % del total para hacer el estudio. En el caso de las mujeres son 56674 que representan el 46 % del total.

Si no distinguimos por sexos el 97 % de la población aragonesa de entre 65-75 años padece una o más enfermedades de la clausura, mientras que el 85 % padece dos o más.

En consecuencia, resulta de gran interés el estudio conjunto de las patologías de los patrones escogidos.

En la figura 3.9 se observan las probabilidades absolutas en el caso de hombres, representándose los arcos con probabilidad superior al 1,1 %. Por ejemplo, a casi el 7 % de los varones aragoneses se les diagnostican problemas metabólico-lipídicos después de hipertensión.Por otro lado, a casi el 4 % se les diagnostica diabetes posteriormente a hipertensión, en torno al 3 % problemas de próstata o también un 3 % cataratas tras hipertensión. Se sigue observando un flujo fuerte entre hipertensión y problemas en el metabolismo, pero también entre diabetes e hipertensión o entre diabetes y problemas metabólicos.

De forma análoga, se observa el grafo de probabilidades absolutas para mujeres en la figura 3.10, representándose los arcos con probabilidad superior al 1 %. El 7 % de las pacientes padece problemas de metabolismo tras ser diagnosticadas con hipertensión, mientras que al 6 % les ocurre a la inversa. En torno al 5 % fueron diagnosticadas con osteoartritis tras la hipertensión y al 3 % les ocurrió a la inversa. Se observa el mayor flujo de pacientes entre hipertensión y metabolismo, hipertensión y osteoartritis. También destacar el flujo entre alteraciones en el carácter o estado de ánimo e hipertensión.

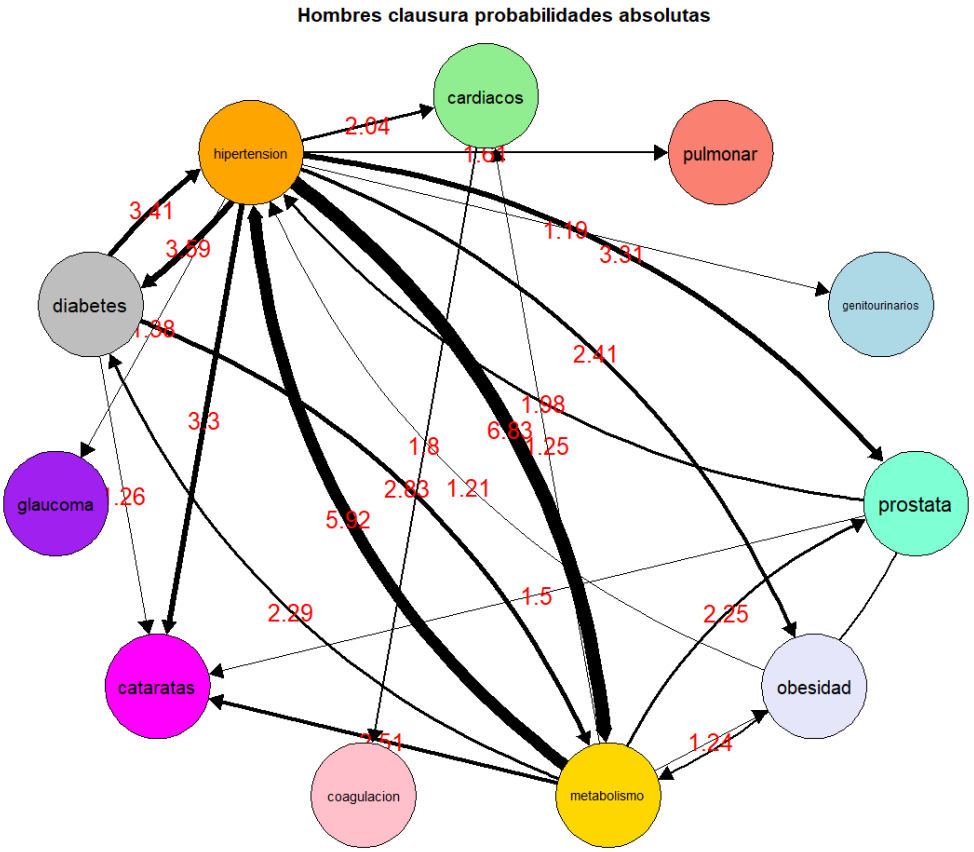


Figura 3.9: Grafo clausura para hombres en probabilidades absolutas, se representan los arcos con probabilidad superior al 1,1 %.

Si consideramos las probabilidades de transición asociadas a la clausura, conviene representarlos con un heatmap, dado que se trata de la matriz de Markov de primer orden y resulta una forma muy visual de ver las probabilidades de paso de una enfermedad a otra. Según el gradiente de colores cuanto

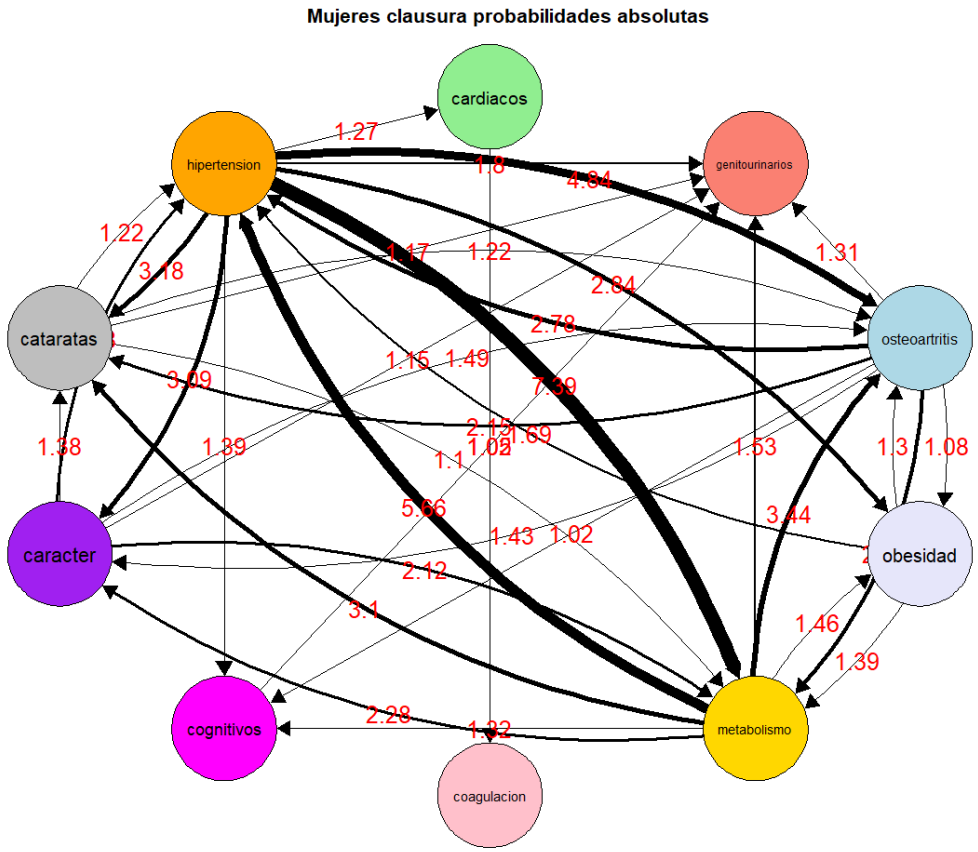


Figura 3.10: Grafo clausura para mujeres en probabilidades absolutas, se representan los arcos con probabilidad superior al 1 %.

más intenso es el color naranja mayor probabilidad.

En la matriz asociada a mujeres de la figura 3.11, se observa que las probabilidades más altas son padecer un problema de coagulación si previamente se ha detectado uno cardiaco. Por otro lado, padecer enfermedades genitourinarias cuando previamente se diagnóstico alteraciones a nivel cognitivo (demencias). Por último la doble interacción hipertensión y trastornos de metabolismo a nivel lipídico que hemos comentado anteriormente. También a nivel de conjunto, se observa que osteoartritis, cataratas, hipertensión y metabolismo tienen una probabilidad muy alta de diagnosticarse después de cualquier enfermedad de la clausura. Remarcar el caso de la obesidad, en el que se ve que principalmente es causa y no consecuencia de patologías, tiene altas probabilidades en derivar en problemas de hipertensión, problemas metabólicos u osteoartritis.

En el heatmap de la figura 3.12 asociado a hombres análogamente al caso femenino existe una probabilidad muy alta de que las arritmias (cardiacos) derive en alteraciones en la coagulación o que tanto el glaucoma tiene probabilidad alta en que la siguiente enfermedad en detectarse sea la hipertensión. Del mismo modo una diabetes es el antecedente de diagnosticar alteraciones metabólicas. Por supuesto, hipertensión y metabolismo que se dan con alta probabilidad en ambos sentidos. También se puede observar que cataratas, coagulación o problemas genitourinarios tiene alta probabilidad de derivar en arritmias. Del mismo modo, padecer hipertensión o problemas metabólicos es muy probable si se padece obesidad, diabetes, glaucoma, problemas de próstata, EPOC (pulmonares) o incluso cataratas.

Conviene remarcar que las matrices obtenidas son asimétricas, por ejemplo, la probabilidad de pasar

de obesidad a hipertensión no es la misma que en el sentido contrario como se puede observar. Esto se generaliza a todas las casuísticas.

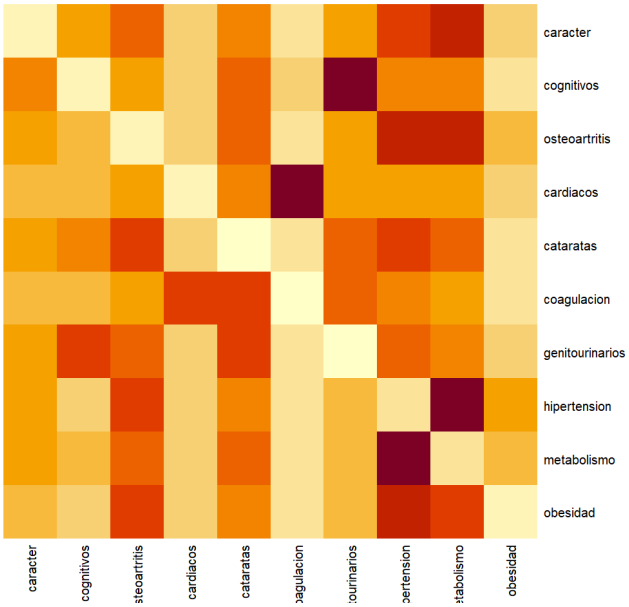


Figura 3.11: Heatmap para matriz de transición de probabilidades de primer orden en el caso de mujeres.

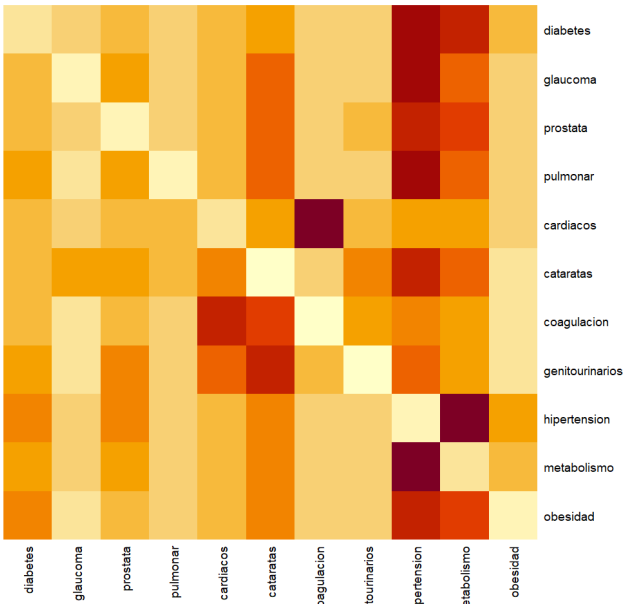


Figura 3.12: Heatmap para matriz de transición de probabilidades de primer orden en el caso de hombres.

3.3. Conclusiones

A través de las técnicas implementadas se han obtenido resultados que nos han permitido obtener una serie de patrones de multimorbilidad y establecer una secuencia en su diagnóstico. En cada patrón aparecen enfermedades que clínicamente están relacionadas entre sí etiológicamente, con otras provenientes de otras causas distinta. Esta agrupación nos la permite hacer el análisis clúster. De esta forma,

resulta más sencillo establecer no solo la secuencia de diagnóstico, con patologías concretas, sino aportar información sobre las enfermedades en que podría derivar una trayectoria mediante el estudio de las enfermedades que figuran en el mismo clúster que las patologías de la trayectoria escogida.

Para ejemplificar lo anterior, considerar el patrón relativo a hombres,

{epoc, metabólico-lipídicos, hipertensión, coagulación, arritmias}

La regla de asociación nos indica que un paciente con las primeras cuatro enfermedades deriva con una confianza muy alta en padecer arritmias. Además, la secuencia de diagnóstico más probable en este caso sería,

{ hipertensión, metabólico-lipídicos, epoc, arritmias, coagulación}

En este ejemplo, tenemos enfermedades de tres clusters. Un paciente con hipertensión y metabólicos, enfermedades del clúster 4 puede derivar en obesidad, diabetes, problemas de tiroides o incluso neoplasias. Las EPOC, pueden derivar en otras enfermedades repiratorias del clúster 2, en problemas de próstata o incluso en ansiedad o migraña. El último clúster estaría compuesto por arritmias y problemas de coagulación que suelen aparecer conjuntamente.

Como trabajo futuro y con el objetivo de ampliar el estudio, siguiendo la metodología de [19], [15], [18], podría ser interesante disminuir el umbral de prevalencia, de cara a aumentar el numero de patologías a estudio y aportar mayor riqueza y variabilidad tanto en la parte de clusters como en las trayectorias de multimorbilidad.

Se puede concluir que el estudio proporciona un mecanismo de ayuda en la anticipación en el diagnóstico, de cara a prevenir en la medida de lo posible la aparición de enfermedades crónicas propias de la edad de 65 a 75 años dentro de la población de Aragón. Esto es de interés para la planificación y gestión de recursos sanitarios actuales y en políticas de prevención futuras.

Bibliografía

[1] Agrawal R, Srikant R. Fast algorithms formining association rules in large databases.In Proc.20th Int. Conf. on Very Large DataBases, 487–499. 1994.

[2] Campello R, Hruschka E, A fuzzy extension of the silhouette width criterion for cluster analysis, Fuzzy Sets and Systems, Volume 157, Issue 21, pages 2858-2875, 2006.

[3] Choi SHC, Cha S, Tappert C. A Survey of Binary Similarity and Distance Measures. J. Syst. Cybern. Inf. 8. 2009

[4] Hahsler M, Grün B, Hornik K. arules - A Computational Environment for Mining Association Rules and Frequent Item Sets. Journal of Statistical Software, 14(15), 1–25, 2005.

[5] Hastie T, Tibshirani R, Friedman J. Elements of Statistical Learning. Springer Series in Statistics Springer New York Inc., New York, NY, USA, 2001.

[6] Sun K, Bai F. Mining Weighted Association Rules without Preassigned Weights, IEEE Transactions On Knowledge And Data Engineering", volume.20,issue 4, 2008.

[7] Kumar Dr, Rukmani K. Mining weighted association rule using HITS, 2010.

[8] Pei J, Han J, Mao R. CLOSET: An Efficient Algorithm for Mining Frequent Closed Itemsets, 2001.

[9] Prados-Torres A, Poblador-Plou B, Gimeno-Miguel A, Calderón-Larrañaga A, Poncel-Falcó A, Gimeno-Feliú L A , González-Rubio F, Laguna-Berna C , Marta-Moreno J, Clerencia-Sierra M, Aza-Pascual-Salcedo M, Bandrés-Liso A C, Coscollar-Santaliestra C, Pico-Soler V , Abad-Díez J M, Cohort Profile: The Epidemiology of Chronic Diseases and Multimorbidity. The EpiChron Cohort Study, International Journal of Epidemiology, Volume 47, Issue 2, 2018, https://doi.org/10.1093/ije/dyx259.

[10] Ramkumar G, Ranka S, y Tsur S. Weighted Association Rules: Model and Algorithm, 2001.

[11] Ramze Rezaee M, Lelieveldt B, Reiber J, A new cluster validity index for the fuzzy c-mean, Pattern Recognition Letters, Volume 19, Issues 3–4, 1998.

[12] Rawashdeh M, Ralescu A. Fuzzy cluster validity with generalized silhouettes. 841. 11-18, 2012.

[13] Meyn S, Tweedie R . Markov Chains and Stochastic Stability. Springer, 1993.

[14] Serafini A, Giordani P, Ferraro M. fclust: An R Package for Fuzzy Clustering. The R Journal. 9. 10.32614/RJ-2019-017, 2019.

[15] Shi X, Nikolic G, Van Pottelbergh G, van den Akker M, Vos R, De Moor B. Development of Multimorbidity Over Time: An Analysis of Belgium Primary Care Data Using Markov Chains and Weighted Association Rule Mining. J Gerontol A Biol Sci Med Sci. 2021.

[16] Subbalakshmi C, Rama Krishna G, Krishna Mohan Rao S, Venketeswa Rao P. A Method to Find Optimum Number of Clusters Based on Fuzzy Silhouette on Dynamic Data Set. Procedia Computer Science, Volume 46, 2015.

[17] Tan P, Steinbach M, Kumar V. Introduction to Data Mining. Addison Wesley, US ed edition, 2005.

[18] Vetrano D, Roso-Llorach A, Fernández S et al. Twelve-year clinical trajectories of multimorbidity in a population of older adults. Nat Commun 11, 3223, 2020.

[19] Violán C, Fernández-Bertolín S, Guisado-Clavero M, Foguet-Boreu Q, Valderas JM, Vidal Manzano J, Roso-Llorach A, Cabrera-Bean M. Five-year trajectories of multimorbidity patterns in an elderly Mediterranean population using Hidden Markov Models. Sci Rep. 2020.

[20] `https://cran.r-project.org/web/packages/fclust/fclust.pdf`

[21] `https://cran.r-project.org/web/packages/arules/arules.pdf`

[22] https://www.sanidad.gob.es/estadEstudios/estadisticas/docs/Manual_de_codificacion.pdf

Anexos

Codificación enfermedades según sistema internacional CIE-10

A continuación se asocian los códigos de las 25 enfermedades que aparecen en el estudio, recogidos en [22], junto con su denominación en la memoria.

- 11-47: neoplasias
- 48: tiroides
- 49-50: diabetes
- 53: metabolismo-lipídicos
- 62: coagulación
- 651: ansiedad
- 653: desórdenes cognitivos/demencia
- 657: desórdenes de carácter
- 84: migraña
- 86: cataratas
- 88: glaucoma
- 89: ceguera
- 94: sordera
- 98-99: hipertensión
- 106: arritmias
- 127: epoc
- 128: asma
- 134: sinusitis
- 163: genitourinarios
- 164: próstata
- 173: desórdenes menopausia
- 203: osteoartritis

- 205: problemas vertebrales
- 206: osteoporosis
- 300: obesidad

Algunos scripts asociados a los resultados del Capítulo 3

```
##############################################################################
######################## GENERACIÓN REGLAS DE ASOCIACIÓN #####################
##############################################################################

library(arules)
library(dplyr)

setwd("C:/Users/ecamp/OneDrive/Escritorio/TFM")

data <- read.transactions("./sexo_1.csv", format = "single", sep= ',',cols = c(1,10))
as(data, "list")

#inspeccion de la composicion de transacciones
inspect(head(data))
inspect(tail(data))


#algoritmo hits para calcular los pesos de ponderación de cada itemset
w<-hits(data,type="relative")

#añadir los pesos a la informacion de las transacciones
transactionInfo(data)[["weight"]] <- w
transactionInfo(head(data))

#algoritmo ECLAT (con ponderacion w)  generación de itemsets frecuentes.

frec<-weclat(data, parameter = list(support = 0.01, confidence= .5,
minlen=1, maxlen=5),
 control = list(verbose = TRUE))

#soporte de itemsets más frecuentes.
inspect(sort(frec))

#Generar reglas de asociacion
rule <- ruleInduction(frec, confidence= .5)
rules.sorted<-sort(rule,by = "lift")
inspect(rules.sorted)

#poda de reglas de asociación redundantes pruning redundant rules
```

```
subset.matrix <- is.subset(rules.sorted,rules.sorted)
subset.matrix[lower.tri(subset.matrix, diag=T)] <- F
redundant <- apply(subset.matrix, 2, any)
rules.pruned <- rules.sorted[!redundant]
inspect(rules.pruned)


##############################################################################
################ GENERACIÓN DE SECUENCIA PARA REGLA ELEGIDA EN MUJERES ###########
##############################################################################

library(dplyr)
library(tibble)
library(lubridate)
library(reshape)
library("qgraph")

setwd("C:/Users/ecamp/OneDrive/Escritorio/TFM")
df<-read.csv("./df_prev_bien.csv",header=TRUE,sep=",")

#filtrado fichero con enfermedades asociadas a la regla
df_seq_mujeres<-df%>%filter(sexo==1)
df_seq_mujeres<-df_seq_mujeres %>%
  filter(enfermedad_agrupadas == "hipertension"|
           enfermedad_agrupadas=="106"|enfermedad_agrupadas=="163"
         |enfermedad_agrupadas=="203"|
         enfermedad_agrupadas=="62")
#data.table::fwrite(df_seq_mujeres, "df_mujeres_frec_rule.csv", sep=',')
#df_seq_mujeres<-read.csv("./df_mujeres_frec_rule.csv",header=TRUE,sep=",")

#establecer rangos numericos por fecha de detección de enfemedad
p1<-df_seq_mujeres%>%group_by(usuario)%>% mutate(numbering = row_number())
p1_filtr<-p1%>% select(usuario, enfermedad_agrupadas,numbering)

#escribirlo en formato horizontal con cast
secuencias<-cast(p1_filtr, usuario ~ enfermedad_agrupadas,value="numbering",
fun.aggregate = min)
#data.table::fwrite(secuencias, "secuencias.csv", sep=',')

secuencias[sapply(secuencias, is.infinite)] <- NA
#eliminar pacientes que solo padecen 1 enfermedad de la regla
solouna<-apply(secuencias[,2:6], 1, function(x) sum(!is.na(x)) == 1 )
seq.true<-secuencias[!solouna,]
dim(seq.true)
#obtengo estadístico media de rangos para cada enfermedad de la regla
apply(seq.true[,2:6],2,mean, na.rm=TRUE)

colnames(seq.true)<-c('usuario','cardiacos','genitourinarios','osteoartritis',
'coagulacion','hipertension')
```

```
#############################################################################
############## INTERVALOS BOOOTSTRAP PARA MUJERES ###########################
#############################################################################

#calculo intervalo bootstrap asociado a los rangos medios del patrón/ trayectoria
# de multimorbilidad más frecuente en mujeres

df.seq<-seq.true
df.seq$usuario<-as.factor(df.seq$usuario)
nseq<-ncol(df.seq) # número de elementos en la regla+1
B<-1000 ## número de réplicas

#almacenamos los rangos medios (u otro estadístico) para cada remuestra
reglaranks<-matrix(0, nrow=B, ncol= nseq-1)

for(ib in 1:B){
  Bpatiens<-sample(df.seq$usuario, size=nrow(df.seq),replace=TRUE)
  reglaranks[ib,]<-apply(seq.true[Bpatiens,2:nseq],2,mean, na.rm=TRUE)
print(ib)
  }

## escribimos los nombres/códigos, el valor medio y los percentiles bootstrap
## que queramos calcular, p.ej. c(0.025, 0.5, 0.975)
rbind(
  colnames(df.seq)[2:nseq],
  apply(seq.true[,2:nseq],2,mean, na.rm=TRUE),
  apply(reglaranks,2,quantile,probs=c(0.025,0.5, 0.975) ))


#############################################################################################
########### GENERACIÓN DE GRAFOS CON PESOS ASOCIADOS A  REGLA ELEGIDA ############
#############################################################################################

namesnodes<-names(seq.true)[-1]
nedges<-length(names(seq.true)[-1])

#genero grafo de pesos nulos
Edges <- data.frame(
  from = rep(namesnodes, each=nedges),
  to = rep(namesnodes, times=nedges),
  thickness = 0) #genero la red/grafo con pesos nulos

#asigno los pesos a cada arco en funcion de si tiene posición en la secuencia
#de enfermedades
for (ipatiens in 1: nrow(seq.true)){
  names_comrb<-names(seq.true)[which(!is.na(seq.true[ipatiens,]))[-1]]
  as.factor(names_comrb)
  order_comrb<-order(seq.true[ipatiens,names_comrb])
  path_comrb<-names_comrb[order_comrb]
  npath<-length(path_comrb)
  for (i in 1:(npath-1)){
```

```
    arco<- which((Edges$from == path_comrb[i]) & (Edges$to == path_comrb[i+1]))
    Edges$thickness[arco] = Edges$thickness[arco]+1
  }
}

#elimino las aristas que son bucle
Edges <- subset(Edges,from!=to)

data.table::fwrite(Edges, "mujeres_frec_rule.csv", sep=',')


#obtengo a partir de los pesos probabilidades absolutas y relativas
#genero los .csv que se leen a continuación

## Representación grafo probabilidades relativas mujeres para la regla frecuente
prob_transicion_rel <- read.csv2("C:/Users/ecampol/Desktop/grafos
/mujeres_frec_rule_rel.csv")
Edges_rel <- data.frame(
  from = prob_transicion_rel$from,
  to = prob_transicion_rel$to,
  thickness = prob_transicion_rel$prob_rel)

qgraph(Edges_rel,layout = "spring",directed=TRUE,esize=20,edge.labels=T,label.prop=3,
label.cex=0.65,vsize=15, color =c("LightGreen","Salmon","LightBlue","Lavender","Gold",
"Pink",
"Magenta","Purple", "Grey", "Orange"), edge.color = "black", edge.label.cex=2.5,
fade=FALSE,
 edge.label.bg=FALSE, edge.label.color="red",label.color = "black",minimum=14)
title("Mujeres regla frecuente probabilidades relativas", line = 2.5)

summary(Edges_rel$thickness)

## Representación grafo probabilidades absolutas mujeres para la regla frecuente
prob_transicion_abs <- read.csv2("C:/Users/ecampol/Desktop/grafos/
mujeres_frec_rule_abs.csv")
Edges_abs <- data.frame(
  from = prob_transicion_abs$from,
  to = prob_transicion_abs$to,
  thickness = prob_transicion_abs$prob_abs)

qgraph(Edges_abs,layout = "groups",directed=TRUE,esize=20,edge.labels=T,label.prop=3,
label.cex=0.65,vsize=15, color =c("LightGreen","Salmon","LightBlue","Lavender","Gold",
"Pink","Magenta","Purple", "Grey", "Orange"), edge.color = "black", edge.label.bg=FALSE,
edge.label.cex=2.5, fade=FALSE, edge.label.color="red",
label.color = "black",minimum=1.6)
title("Mujeres regla frecuente absolutas", line = 2.5)
summary(Edges_abs$thickness)


########################################################################
############### CONSTRUCCION GRAFO TRAYECTORIAS ########################
```

```
#########################################################################


#representación grafica con la secuencia de los 5 patrones obtenidos
#para hombres y 5 patrones obtenidos para mujeres

library("igraph")
g_hombres<-read.csv("secuencias_hombres.csv", sep = ";")
g_mujeres<-read.csv("secuencias_mujeres.csv", sep = ";")
g_hombres <- graph.data.frame(g_hombres, directed = TRUE)
g_mujeres <- graph.data.frame(g_mujeres, directed = TRUE)

V(g_hombres)$color<-"lightblue"
tkplot(g_hombres,sep = "")
V(g_mujeres)$color<-"orange"
tkplot(g_mujeres,sep = "")


#########################################################################################
#### GENERACIÓN SECUENCIAS CON ENFERMEDADES DE LA CLAUSURA PARA MUJERES #########
#########################################################################################

library(dplyr)
library(tibble)
library(lubridate)
library(reshape)
library("qgraph")


setwd("C:/Users/ecamp/OneDrive/Escritorio/TFM")
df<-read.csv("./df_prev_bien.csv",header=TRUE,sep=",")

#filtrado fichero con enfermedades de la clausura
df_seq_mujeres<-df%>%filter(sexo==1)
df_seq_mujeres<-df_seq_mujeres %>%
  filter(enfermedad_agrupadas == "hipertension"|
         enfermedad_agrupadas=="106"|enfermedad_agrupadas=="163"
         |enfermedad_agrupadas=="653"|enfermedad_agrupadas=="203"|
         enfermedad_agrupadas=="657"
         |enfermedad_agrupadas=="62"|enfermedad_agrupadas=="86"|
         enfermedad_agrupadas=="53"|enfermedad_agrupadas=="300")
#data.table::fwrite(df_seq_mujeres, "df_seq_mujeres.csv", sep=',')
#df_seq_mujeres<-read.csv("./df_seq_mujeres.csv",header=TRUE,sep=",")

#establecer rangos numericos por fecha de detección de enfemedad
p1<-df_seq_mujeres%>%group_by(usuario)%>% mutate(numbering = row_number())
p1_filtr<-p1%>% select(usuario, enfermedad_agrupadas,numbering)

#escribirlo en formato horizontal con cast
secuencias<-cast(p1_filtr, usuario ~ enfermedad_agrupadas,value="numbering",
fun.aggregate = min)
```

```
#data.table::fwrite(secuencias, "secuencias.csv", sep=',')

#elimino los pacientes con solo una morbilidad en la clausura
secuencias[sapply(secuencias, is.infinite)] <- NA
solouna<-apply(secuencias[,2:11], 1, function(x) sum(!is.na(x)) == 1 )
seq.true<-secuencias[!solouna,]
dim(seq.true)
#obtengo estadístico media de rangos
apply(seq.true[,2:11],2,mean, na.rm=TRUE)

colnames(seq.true)<-c('usuario','cardiacos','genitourinarios','osteoartritis',
'obesidad', 'metabolismo', 'coagulacion','cognitivos','caracter',
'cataratas','hipertension')


##############################################################################
##### GENERACIÓN DE GRAFOS CON PESOS ASOCIADOS A ENFERMEDADES DE LA CLAUSURA ########
##############################################################################

#vector cadena caracteres
namesnodes<-names(seq.true)[-1]
nedges<-length(names(seq.true)[-1])

#genero el grafo con pesos nulos
Edges <- data.frame(
  from = rep(namesnodes, each=nedges),
  to = rep(namesnodes, times=nedges),
  thickness = 0)

#asigno los pesos a cada arco en funcion de si tiene posición en la secuencia
de enfermedades

for (ipatiens in 1: nrow(seq.true)){
  names_comrb<-names(seq.true)[which(!is.na(seq.true[ipatiens,]))[-1]]
  as.factor(names_comrb)
  order_comrb<-order(seq.true[ipatiens,names_comrb])
  path_comrb<-names_comrb[order_comrb]
  npath<-length(path_comrb)
  for (i in 1:(npath-1)){
    arco<- which((Edges$from == path_comrb[i]) & (Edges$to == path_comrb[i+1]))
    Edges$thickness[arco] = Edges$thickness[arco]+1
  }
}

#elimino las aristas que son bucle
Edges <- subset(Edges,from!=to)

data.table::fwrite(Edges, "prob_mujeres.csv", sep=',')

#obtengo a partir de los pesos probabilidades absolutas y relativas
#genero los .csv que se leen a continuación
```

```
## Representación grafo probabilidades absolutas mujeres
prob_transicion_abs <- read.csv2("C:/Users/ecamp/OneDrive/Escritorio/
TFM/prob_mujeres_abs.csv")

Edges_abs <- data.frame(
  from = prob_transicion_abs$from,
  to = prob_transicion_abs$to,
  thickness = prob_transicion_abs$prob_abs)

qgraph(Edges_abs,layout = "spring",directed=TRUE,esize=20,edge.labels=T,label.prop=3,
       color =c("LightGreen","Salmon","LightBlue","Lavender","Gold","Pink","Magenta",
"Purple", "Grey", "Orange"),edge.color = "black", edge.label.bg=FALSE, edge.label.cex=3,
fade=FALSE,edge.label.color="red",label.color = "black",minimum=1.0)
title("Mujeres clausura absolutas", line = 2.5)

summary(Edges_abs$thickness)

## Representación grafo probabilidades relativas para mujeres
prob_transicion_rel <- read.csv2("C:/Users/ecamp/OneDrive/Escritorio/TFM/
prob_mujeres_rel.csv")
Edges_rel <- data.frame(
  from = prob_transicion_rel$from,
  to = prob_transicion_rel$to,
  thickness = prob_transicion_rel$prob_rel)

#libreria qgraph
qgraph(Edges_rel,layout = "groups",directed=TRUE,esize=20,edge.labels=T,
label.prop=3,
       color =c("LightGreen","Salmon","LightBlue","Lavender","Gold","Pink",
       "Magenta","Purple", "Grey", "Orange"), edge.color = "black", edge.label.bg=FALSE,
edge.label.cex=3, fade=FALSE, edge.label.color="red",label.color = "black",minimum=12)
title("Mujeres clausura relativas", line = 2.5)

summary(Edges_rel$thickness)

##############################################################################
###################### HEATMAP PARA HOMBRES CON PROBABILIDADES RELATIVAS #############
################## MATRIZ DE TRANSICIÓN DE PROBABILIDADES DE PRIMER ORDEN ##########
##############################################################################

prob_transicion_rel <- read.csv2("C:/Users/ecamp/OneDrive/Escritorio/
TFM/prob_hombres_rel.csv")
Edges_rel <- data.frame(
  from = prob_transicion_rel$from,
  to = prob_transicion_rel$to,
  thickness = prob_transicion_rel$prob_rel)

#Edges$ptotal<- Edges$thickness/sum(Edges$thickness)
#Edges$pcond<- Edges$thickness/rep(xtabs(Edges$thickness~Edges$from),
```

```
each=nlevels(as.factor(Edges$from))-1)


Edges_rel$from<-as.factor(Edges_rel$from)
Edges_rel$to<-as.factor(Edges_rel$to)


## Conversión de Edges a una matriz válida para el heatmap

nnodos<-nlevels(as.factor(Edges_rel$from))
mheat<-matrix(0,nrow=nnodos,ncol=nnodos)
for ( i in 1:nrow(Edges_rel)){
  mheat[as.numeric(Edges_rel$from[i]),
  as.numeric(Edges_rel$to[i])]<-Edges_rel$thickness[i]
}
rownames(mheat)<-levels(as.factor(Edges_rel$from))
colnames(mheat)<-rownames(mheat)


#ordeno enfermedades en no comunes y comunes por sexo para establecer comparativa
order<-c('diabetes','glaucoma','prostata','pulmonar','cardiacos', 'cataratas',
'coagulacion', 'genitourinarios','hipertension','metabolismo', 'obesidad')
mheat<-mheat[order,order]
heatmap(mheat, Colv=NA, Rowv=NA, revC=TRUE)

################################################################################
###################### OBTENCIÓN CLUSTERS FUZZY ################################
######################### CASO MUJERES #########################################
################################################################################
library(factoextra)
library(dplyr)

#lectura
setwd("C:/Users/ecamp/OneDrive/Escritorio/TFM")
primer_cluster<-read.csv("./matriz_primer_cluster.csv",header=TRUE,sep=",")
#adapto el data frame
primer_cluster_mujeres<-primer_cluster%>%filter(sexo==1)
primer_cluster<-primer_cluster_mujeres[,c(1:26)]
colnames(primer_cluster)<-c("usuario","arritmias","epoc","asma","sinusitis",
                            "genitourinarios","menopausia","osteoartritis",
                               "vertebral","osteoporosis","obesidad","tiroides",
                               "metabolismo","coagulación","ansiedad","demencia",
                               "caracter","migraña", "cataratas","glaucoma","ceguera"
                               ,"sordera","diabetes",
                               "hipertension","neoplasias")
primer_cluster_redd<-primer_cluster[,1:26]
primer_cluster_redd$diabetes[which(primer_cluster_redd$diabetes > 1)]<-1
primer_cluster_redd$hipertension[which(primer_cluster_redd$hipertension > 1)]<-
1
primer_cluster_redd$neoplasias[which(primer_cluster_redd$neoplasias > 1)]<-1
summary(primer_cluster_redd)
```

```
primer_cluster_fac<-primer_cluster_redd
for (i in 2:ncol(primer_cluster_redd)){
  primer_cluster_fac[,i]<-factor(primer_cluster_redd[,i])
}
summary(primer_cluster_fac)

#hard-clustering con metodo jerárquico.
tree1<-hclustvar(X.quali=primer_cluster_fac[,-1])
plot(tree1)
set.seed(1234)
estabilidad<-stability(tree1, B=10)
kopt<- 4
summary(cutreevar(tree1,k=kopt, matsim=TRUE))

#revisión de métricas.
D_primer_1<-
  proxy::as.dist(
    simil(primer_cluster_redd[,-1], by_rows = FALSE, method="jaccard"))
# D_primer_c<-smacof::sim2diss(D_primer_c) # solo si necesitamos ver la matriz
D_primer_2 <-
  proxy::as.dist(
    simil(primer_cluster_redd[,-1], by_rows = FALSE, method="Kulczynski2"))
D_primer_3 <-
  proxy::as.dist(
    simil(primer_cluster_redd[,-1], by_rows = FALSE, method="ochiai"))
D_primer_4 <-
  proxy::as.dist(
    simil(primer_cluster_redd[,-1], by_rows = FALSE, method="yule"))
D_primer_5 <-
  proxy::as.dist(
    simil(primer_cluster_redd[,-1], by_rows = FALSE, method="yule2"))
D_primer_6 <-
  proxy::as.dist(
    simil(primer_cluster_redd[,-1], by_rows = FALSE, method="Simpson"))
D_primer_7 <-
  proxy::as.dist(
    simil(primer_cluster_redd[,-1], by_rows = FALSE, method="Braun-Blanquet"))

#elegir una métrica.
plot(D_primer_2,D_primer_6) #observar la correlación pero no la coincidencia
abline(0,1)
var(D_primer_1)
var(D_primer_2) ## es bastante más variable
var(D_primer_3)
var(D_primer_4)
var(D_primer_5)
var(D_primer_6)
var(D_primer_7)

#asigno métrica
D_primer<-D_primer_2 # asigno la matriz con la métrica 2.
```

```
#fijo semilla
set.seed(1234)
#aplico fuzzy-clustering
NF_primer<- NEFRC(as.dist(D_primer),k=4,m=1.14, RS=10,index="SIL.F")
summary(NF_primer)
plot(NF_primer, ucex=TRUE,v1v2=c(1,3))
```