Revista de Matemática: Teoría y Aplicaciones 2000 **7**(1-2) : 23–41
cimpa – ucr – ccss issn: 1409-2433

ALGORITMOS PARA LA CLASIFICACIÓN PIRAMIDAL SIMBÓLICA

Oldemar Rodríguez [*] – Maria Paula Brito[**] – Edwin Diday[***]

Recibido: 23 Octubre 1999

Resumen

En este artículo se define el concepto de pirḿide simbólica, además se presentan dos algoritmos para generar este tipo de piramide a partir de una matriz de datos simbólicos. El primer algoritmo (CAPS) encuentra un "orden total compatible con la pirámide" de los n objetos, mientras que el segundo (CAPSO) construye la pirámide a partir de un orden dado a priori en los objetos, dicho orden se recibe como entrada en el algoritmo. Ambos algoritmos, además de producir la pirámide, para cada grada encuentran el objeto simbólico asociado a cada nodo y su extensión. También se presentan los teoremas de convergencia.

Palabras clave: pirámide, objeto simbólico, grada, grado de generalidad, objeto completo, componente conexa, tablas de datos simbólica.

Abstract

This pyramidal clustering method generalizes hierarchies by allowing non-disjoint classes at a given level instead a partition. Moreover, the clusters of the pyramid are intervals of a total order on the set being clustered, hence pyramids constitute an intermediate model between the tree and the lattice structures. This method allows moreover to cluster more complex data than the tabular model allows to process, by considering variation on the values taken by the variables. Each cluster formed is defined not only by the set of its elements (i.e. its extent) but also by a symbolic object, which describes its properties (its intent). In this paper we propose a new algorithm

[*] CEREMADE, Université de Paris IX – Dauphine, Place du Maréchal de Lattre de Tassigny, 75775 Paris cedex 15 Francia; E-Mail: orodrigu@ceremade.dauphine.fr & CIMPA, Escuela de Matemática, Universidad de Costa Rica, 2060 San José, Costa Rica; E-Mail: orodrigu@cariari.ucr.ac.cr

[**] Faculdade de Economia do Porto, Rua Dr. Roberto Frias, 4200 Porto, Portugal; E-Mail: mpbrito@fep.up.pt

[***] CEREMADE, Université de Paris IX – Dauphine, Place du Maréchal de Lattre de Tassigny, 75775 Pais cedex 15 Francia; E-Mail: diday@ceremade.dauphine.fr

CAPS to built a symbolic pyramid, this algorithm in an extension to symbolic case of the algorithm CAP proposed in [Diday 1984] to the symbolic case. An example is presented to illustrate the effectiveness of the proposed algorithm and we also present a free software for this algorithm.

Keywords: symbolic data analysis, pyramidal clustering, exten, inten, conceptual lattices symbolic pyramid

Mathematics Subject Classification: 62H30, 68T10

1. Objetos simbólicos

Antes de definir formalmente un objeto simbólico se presenta la notación necesaria, sean:

- Ω el conjunto de individuos.
- O_j el espacio de descripción para la variable j.
- $P(O_j)$ el conjunto de partes de O_j.
- La descripción de un individuo ω está representada por el vector $(y_1(\omega), \ldots, y_p(\omega))$ donde cada variable y_j, $j = 1, 2, \ldots, p$ es una aplicación de Ω en $P(O_j)$. El valor de $y_j(\omega)$ puede estar representado por un conjunto de valores, un intervalo o bien un histograma, entre otros.
- Sea $D = P(O_1) \times P(O_2) \times \cdots \times P(O_p)$ el conjunto de las posibles descripciones y $d \in D$ una descripción, de modo que para todo $j = 1, 2, \ldots, p$, d_j representa una descripción como un conjunto de valores.

En [9, Diday (1999)] se presenta la siguiente definición de Objeto Simbólico:

Definición 1 Un objeto simbólico es un triplete (a, R, d) donde R en un vector de relaciones R_i, $d = (d_1, d_2, \ldots, d_p)$ es un vector de descripciones d_i, y a es una aplicación de Ω en $\{T, F\}$.

Si en la definición anterior tomamos $a(w) = [y_1(w)R_1d_1] \wedge [y_2(w)R_2d_2] \wedge \cdots \wedge [y_p(w)R_pd_p]$ donde $a(w) = T$ si y solo $y_j(w)R_jd_j$ para todo $j = 1, 2, \ldots, p$ entonces el objeto simbólico se conoce como Objeto de Aserción.

Si $[y_j(w)R_jd_j] \in L = \{T, F\}$ para todo $j = 1, 2, \ldots, p$ el objeto simbólico se conoce como *Objeto Booleano* y si $[y_j(w)R_jd_j] \in L = [0, 1]$ para todo $j = 1, 2, \ldots, p$ el objeto simbólico se conoce como *Objeto Modal*.

En el caso de objetos booleanos la extensión de define por $ext_\Omega(a) = \{w \in \Omega$ tal que $a(w) = T\}$; mientras que en el caso de objetos simbólicos modales la extensión de a de nivel α se define por $ext_\Omega(a, \alpha) = \{w \in \Omega$ tal que $a(w) \geq \alpha\}$.

Definición 2 (Orden simbólico) Sea S el conjunto de objetos simbólicos definidos sobre las mismas variables, entonces $\forall\ s_1, s_2 \in S$ se dice que:

$$s_1 \leq s_2 \Longleftrightarrow ext_\Omega s_1 \subseteq ext_\Omega s_2$$

La relación $\leq$ induce un pre–orden parcial llamado *Pre–orden Simbólic*o [7, Diday (1987)].

Definición 3 (Herencia entre objetos simbólicos) $\forall\ s_1, s_2 \in S$ se dice que s_1 hereda de s_2 si y solo si $s_1 \leq s_2$. Se dirá que s_2 es más general que s_1 y que s_1 es más específico que s_2.

Para la construcción de Pirámides Simbólicas (sección 2.2) será necesario calcular la unión y la intersección entre objetos simbólicos, esta operaciones se definen como sigue [7, Diday (1987)].

Definición 4 Sea $s_1 = (a_{1,}R, d_1)$ y $s_2 = (a_{2,}R, d_2)$ dos objetos simbólicos, la unión entre s_1 y s_2 se denota por $s_1 \cup s_2$, se define como la conjunción de todos los objetos simbólicos, tal que su extensión sobre Ω contiene a $ext_\Omega s_1 \cup ext_\Omega s_2$, es decir, la unión de todos los objetos simbólicos e_i tal que para todo i se tiene que $ext_\Omega(e_i) \supseteq ext_\Omega s_1 \cup ext_\Omega s_2$. Analogamente se define la intersección entre s_1 y s_2 como la conjunción de todos los objetos simbólicos, tal que su extensión sobre Ω contiene a $ext_\Omega s_1 \cap ext_\Omega s_2$.

Resulta claro que si $s_1 = [y_1 \in V_1] \wedge \cdots \wedge [y_p \in V_p]$ y $s_2 = [y_1 \in W_1] \wedge \cdots \wedge [y_p \in W_p]$ entonces $s_1 \cup s_2 = [y_1 \in V_1 \cup W_1] \wedge \cdots \wedge [y_p \in V_p \cup W_p]$ y $s_1 \cap s_2 = [y \in V_1 \cap W_2] \wedge \cdots \wedge [y_p \in V_p \cap W_p]$.

Un concepto importante dentro de la clasificación piramidal simbólica, es la completitud del objeto simbólico. Se dice que un objeto simbólico es completo si este describe de manera exhaustiva ("completa") a su extensión, formalmente:

Definición 5 [3, Brito (1991)] Sean S el conjunto de todos los objetos de aserción, $A = \{a_{1,}a_2, \ldots, a_n\} \subseteq S$, $f : S \to P(A)$ tal que $f(a) = ext_A(a)$ y $g : P(A) \to S$ tal que $\forall\ P \in P(A)$, $P \subseteq f \circ g(P)$. Se denota por $h = g \circ f$. Entonces se dice que el objeto simbólico a es completo si y solo si $h(a) = a$. h se denomina opearador de completitud.

Ejemplo 1 Sea $f : S \to P(A)$ tal que $f(a) = f(\bigwedge_j [y_j \in W_j]) = \{a_i = \bigwedge_j [y_j \in V_j^i]\ /\ V_j^i \subseteq W_j,\ j = 1, 2, \ldots, p\}$ and $g : P(A) \to S$ tal que $g(\{a_1, \ldots, a_m\}) = \alpha = [y_j = \bigcup_i V_j^i]$ entonces $h = g \circ f$ es un operador de completitud.

El algoritmo de clasificación piramidal simbólica que presentamos en la sección 2.2 tiene dos pasos centrales, el paso de generalización en el cual se debe calcular la unión entre objetos simbólicos y el paso de agregación en el que se calcula el "Grado de Generalidad" del objeto simbólico. Se presenta una definición del "Grado de Generalidad" basados en la definición dada por Brito en [4, Brito (1998)] que nos permitirá calcular este índice aún cuando la matriz de datos simbólicos tenga al mismo tiempo variables de tipo intervalo, cuantitativa discreta o de tipo de tipo histograma.

Definición 6 Sea $s = \bigwedge_{j=1}^{p} e_j$ un objeto simbólico, se define el Grado de Generalidad de s por:

$$G(s) = \prod_{j=1}^{p} G(e_j)$$

donde

$$G(e_j) = \begin{cases} \frac{|V_j|}{|\mathcal{Y}_j|} & \text{si} \quad e_j = [y_j \in V_j],\ V_j \subseteq \mathcal{Y}_j \text{ con } \mathcal{Y}_j \text{ discreto.} \\ \frac{\text{longitud}(V_j)}{\text{longitud}(\mathcal{Y}_j)} & \text{si} \quad e_j = [y_j \in V_j],\ V_j \subseteq \mathcal{Y}_j \text{ con } \mathcal{Y}_j \text{ continuo.} \\ \frac{\sum_{h=1}^{k} w_h}{k} & \text{si} \quad e_j = [y_j = \{m_1(w_1), \ldots, m_k(w_k)\}] \text{ es una distribución de frecuencia de la variable } y_j \text{ discreta.} \end{cases}$$

Nótese que el grado de generalidad es una función creciente, es decir si $s_1 \leq s_2$ entonces $G(s_1) \leq G(s_2)$. Esta es una observación importante pues implica que la pirámide no tendrá inversiones.

2. Algoritmos de Clasificación Piramidal Simbólica

En esta sección presentamos dos algoritmos para generar una piramide simbólica a partir de una matriz de datos simbólicos. El primer algoritmo encuentra un "orden total compatible con la pirámide" de los n objetos, mientras que el segundo construye la pirámide a partir de un orden dado a priori en los objetos, dicho orden se recibe como entrada en el algoritmo. Ambos algoritmos, además de producir la pirámide, para cada grada encuentran el objeto simbólico y su extensión.

2.1. Definiciones básicas

En esta sección se presentan los conceptos fundamentales que nos permitirán en las siguientes secciones presentar los algoritmos mencionados anteriormente, para mayores detalles se puede consultar [5, Diday (1984)], [2, Bertrand y Diday (1990)] o [11, Mfoumoune(1998)].

Definición 7 Un índice de disimilitud en un conjunto de objetos Ω es una función $d : \Omega \longrightarrow [0, +\infty[$ tal que:

- $d(w_1, w_2) = d(w_2, w_1)\ \forall\ w_1, w_2 \in \Omega$.
- $d(w, w) = 0, \forall\ w \in \Omega$.

Por otra parte, para cuantificar la disimilitud entre grupos de objetos del conjunto a clasificar, se utilizan los índices de agregación o simplemente agregaciones.

Definición 8 Una agregación es una función $\delta : P(\Omega) \times P(\Omega) \longrightarrow [0, +\infty[$ tal que $\delta(C_1, C_2) = \delta(C_2, C_1)$, donde $P(\Omega)$ es el conjunto de partes de Ω no vacías y disjuntas dos a dos.

Para el caso que nos interesa, la clasificación piramidal simbólica, no utilizaremos un índice de disimilitud ni un índice de agregación, pues en la fase de agregación (paso de generalización) del algoritmo (sección 2.2) se toma la unión de los dos objetos simbólicos que forman la nueva grada, generándose de nuevo un objeto simbólico (pues la unión de objetos simbólicos produce un nueve objeto simbólico), luego para calcular la "disimilitud[1]" (o la "agregación") entre este nuevo objeto simbólico y cualquier otro se utiliza el *Grado de Generalidad* (ver definición 6).

Definición 9 Una jerarquía binaria sobre un conjunto de objetos denotado por Ω es una colección H de partes no vacías de Ω, llamadas nodos o clases que poseen las siguientes propiedades:

- $\{s\} \in H$ para todo $s \in \Omega$.
- $\Omega \in H$.
- Para todo $s \in H$ tal que $\text{card}(s) > 1$, existen s_1, $s_2 \in H$ tales que $s = s_1 \cup s_2$ y $s_1 \cap s_2 = \emptyset$. Esto significa que toda parte de la jerarquía H, con más de un elemento, es la unión disjunta de dos partes pertenecientes a H.

Diday generaliza en [5, Diday (1984)] el concepto de jerarquía binaria al de pirámide, como se muestra en las siguientes definiciones.

Definición 10 Sea θ un orden total sobre Ω y P un conjunto de partes no vacias de Ω. Un elemento $h \in P$ se dice conexo según el orden total θ, si para todo $w \in \Omega$ que esté entre el máx(h) y el mín(h) (mín(h)$\theta w \theta$ máx(h)) se tiene que $w \in h$.

Definición 11 Un orden total θ sobre Ω es compatible con P, un conjunto de partes de Ω, si todo elemento $h \in P$ es conexo según el orden total θ.

Definición 12 Sea Ω un conjunto finito, sea P un conjunto de partes no vacias de Ω (llamadas gradas), P es una pirámide si cumple lo siguiente:

1. $\Omega \in P$.
2. $\forall\ w \in \Omega$ se tiene que $\{w\} \in P$ (gradas terminales).
3. $\forall\ (h, h') \in P \times P$ se tiene que $h \cap h' \in P$ o $h \cap h' = \emptyset$.
4. Existe un orden total θ en Ω compatible con P.

Una pirámide en la que cada grada, que no es una grada terminal, está formada por la unión de dos gradas distintas, se llama *Pirámide Binaria*.

Definición 13 Una pirámide indexada es un par (P, f) donde P es una pirámide y f es una función $f : P \to \mathbb{R}^+$ tal que:

[1]Se usan comillas pues el Grado de Generalidad no verifica todos los axiomas de un índice de disimilitud.

- $\forall\ h \in P$ se tiene que $f(h) = 0 \Leftrightarrow h$ es una grada terminal.
- $\forall\ h,\ h^{'} \in P$ se tiene que $h \subset h^{'} \Rightarrow f(h) \leq f(h^{'})$.

Una pirámide se llama indexada en el sentido estricto si $h \subset h^{'} \Rightarrow f(h) < f(h^{'})$, además la pirámide se llama indexada en el sentido amplio si $h \subset h^{'}$ y $f(h) = f(h^{'})$ implica la existencia de $h_1, h_2 \in P$ diferentes de h tal que $h = h_1 \cap h_2$.

Definición 14 Sea Ω un conjunto finito de objetos simbólicos, sea P un conjunto de partes no vacias de Ω (llamadas también gradas), P es una pirámide simbólica si cumple lo siguiente:

1. P es una pirámide.
2. Cada grada de P tiene asociado un objeto simbólico completo.

Definición 15 Un índice de disimilitud pirámidal es un índice de disimilitud d que además verifica lo siguiente:

- $d(s_1, s_2) = 0 \Rightarrow s_1 = s_2$.
- Existe un orden total θ sobre Ω compatible con d, es decir, un orden total θ tal que:

$$s_1 \theta s_2 \theta s_3 \Rightarrow d(s_1, s_3) \geq \text{máx}\{d(s_1, s_2), d(s_2, s_3)\}.$$

2.2. Algoritmos de Clasificación Pirámidal Simbólica

En esta sección se presenta un nuevo algoritmo que construye una pirámide simbólica binaria a partir de una matriz de datos simbólicos.

Diday en [5, Diday (1984)] propone el algoritmo CAP para construir pirámides numéricas, es decir pirámides generadas a partir de una tabla de datos clásica o a partir de una matriz de distancias. También se presentan algoritmos con este propósito en [2, Bertrand y Diday (1990)], [10, Gil (1998)] y [11, Mfoumoune (1998)]. Paula Brito en [3, Brito (1991)] propone un algoritmo que generaliza el algoritmo para construir pirámides numéricas propuesto por Bertrand al caso simbólico. En esta sección se propone un algoritmo diseñado para construir pirámides simbólicas (CAPS), es decir, una pirámide en la que cada nodo es de nuevo un objeto simbólico. Además este algoritmo calcula la extensión de cada uno de estos objetos simbólicos y verifica su completitud.

Seguidamente se presentan las definiciones necesarias para la especificación del algoritmo, estas definiciones difieren un tanto a las definiciones presentadas en ([3, Brito (1991)], [2, Bertrand y Diday (1990)] y [11, Mfoumoune (1998)]), pues todas son locales a la "componente conexa".

Para las siguientes definiciones consideramos un conjunto $\mathcal{P} \subseteq P(\Omega)$ (el conjunto de partes de Ω) que no necesariamente es una pirámide, si no posiblemente es una "pirámide en construcción", por abuso del lenguaje denominaremos como una *grada* a todo elemento de $\mathcal{P}$.

Definición 16 Sea $C \in \mathcal{P}$, C se llama componente conexa si existe un orden total $\leq_C$ asociado a C.

Definición 17 Se dice que una grada $G \in \mathcal{P}$, pertenece a una componente conexa C de $\mathcal{P}$ si $G \subseteq C$. Además diremos que el orden total $\leq_C$ asociado C induce un orden total $\leq_G$ sobre G de la siguiente manera, si $x, y \in G$ entonces $x \leq_G y \Leftrightarrow x \leq_C y$.

Definición 18 Sean G_1 y G_2 gradas de $\mathcal{P}$ se dice que G_1 es interior G_2 si:

- $G_1 \neq G_2$.
- G_1 y G_2 pertenecen a la misma componente conexa C.
- mín$(G_2) <_C$ mín(G_1) y máx$(G_1) <_C$ máx(G_2), donde $\alpha <_C \beta$ significa que $\alpha \leq_C \beta$ y $\alpha \neq \beta$.

Definición 19 Sean G_1 y G_2 gradas de $\mathcal{P}$, se dice que G_1 es sucesor G_2 (G_2 es predecesor G_1) si:

- $G_1 \subset G_2$ en sentido estricto.
- No existe una grada $G \in \mathcal{P}$ tal que $G_1 \subset G \subset G_2$ en sentido estricto.

Definición 20 Una grada $G \in \mathcal{P}$, se llama maximal si no tiene predecesores.

Definición 21 Sean G_1 y G_2 gradas de $\mathcal{P}$ se dice que G_1 está a la izquierda de G_2 (G_2 está a la derecha de G_1) si:

- Pertenecen a la misma componente conexa C.
- mín$(G_1) \leq_C$ mín(G_2) y máx$(G_1) \leq_C$ máx(G_2).

Definición 22 Sean G_1 y G_2 gradas de $\mathcal{P}$, se dice que G_1 está estrictamente a la izquierda de G_2 si:

- Pertenecen a la misma componente conexa C.
- mín$(G_1) <_C$ mín(G_2) y máx$(G_1) =$ máx(G_2).

Definición 23 Sean G_1 y G_2 gradas de $\mathcal{P}$, se dice que G_2 está estrictamente a la derecha de G_1 si:

- Pertenecen a la misma componente conexa C.
- mín$(G_1) =$ mín(G_2) y máx$(G_1) <_C$ máx(G_2).

Definición 24 Sean G_1 y G_2 gradas de $\mathcal{P}$, se dice que G_1 es la grada maximal izquierda de G_2 si:

- G_1 está a la izquierda de G_2.

- G_1 es una grada maximal.
- máx(G_2) = máx(G_1).

Definición 25 Sea G una grada de $\mathcal{P}$ que pertence a la componente conexa C, sean $G_1, G_2, \ldots, G_l$ todas las gradas maximales de la componente conexa C, ordenadas de izquierda a derecha de acuerdo con el orden $\leq_C$ (es decir G_i está a la izquierda de G_{i+1}). Si G_m es la grada maximal izquierda de G y $m < l$ entonces G_{m+1} se llama la grada maximal siguiente de G. Si $m = l$ se dirá que G no tiene grada maximal siguiente.

En la siguiente definición se establecen los criterios para agregación de dos gradas. Cuando ambas gradas pertencen a la misma componente conexa el criterio es básicamente el mismo que el propuesto por Bertrand ([2, Bertrand y Diday (1990)]), sin embargo, en caso de que ambas gradas pertenecen a componentes diferentes se elimina la condición de que la primera grada esté "delante[2]" de la segunda grada. Esto permite construir pirámides más acordes a la estructura de "distancias" entre los objetos (individuos), pues, pedir que la primera grada esté "delante" de la segunda grada provoca que la pirámide final se vea (posiblemente) afectada por el orden inicial y arbitrario de los objetos de Ω. Además en nuestro algoritmo, tal condición no tiene sentido pues no se parte de un orden arbitrario en los objetos, si no más bien de n componentes conexas con un orden lineal trivial[3] asociado a cada una de ellas.

Definición 26 Sean G_1 y G_2 dos gradas de $\mathcal{P}$.

Caso 1: Si G_1 y G_2 pertenecen a la misma componente conexa y denotamos por $\overleftarrow{G}$ a la grada maximal izquierda de G_1 y por $\overrightarrow{G}$ a la grada maximal siguiente de G_1 (si existe[4]). Entonces G_1 y G_2 son agregables si se cumplen las dos condiciones siguientes:

1. G_1 está a la derecha de $\overleftarrow{G}$ y estrictamente a la izquierda de $\overleftarrow{G} \cap \overrightarrow{G}$.
2. G_2 está a la izquierda de $\overrightarrow{G}$ y estrictamente a la derecha de $\overleftarrow{G} \cap \overrightarrow{G}$.

Caso 2: Si G_1 y G_2 NO pertenecen a la misma componente conexa y denotamos por C_1 y C_2 las componentes conexas a las que pertenecen G_1 y G_2 respectivamente. Entonces G_1 y G_2 son agregables si se cumplen las dos condiciones siguientes:

1. mín(G_1) = mín(C_1) o máx(G_1) = máx(C_1).
2. mín(G_2) = mín(C_2) o máx(G_2) = máx(C_2).

Definición 27 Una grada G de $\mathcal{P}$ se llama activa si se cumplen las siguientes tres condiciones:

- Existe otra grada $G^\star$ en $\mathcal{P}$ tal que G es agregable con $G^\star$.

[2]En Brito (1991) se presenta la noción de que una grada esté delante de otra.
[3]El orden es trivial pues cada componente conexa inicial tiene cardinalidad 1.
[4]Si la grada maximal siguiente no existe entonces las gradas no serán agregables.

- $\nexists \widetilde{G} \in \mathcal{P}$ tal que G es grada interior a $\widetilde{G}$.
- G ha sido agregada a lo sumo una vez.

ALGORITMO DE CLASIFICACIÓN ASCENDENTE PIRAMIDAL SIMBÓLICA (CAPS)

Entrada :

- M =Número máximo de iteraciones.
- N =Número de vectores de datos simbólicos (número de la filas de tabla de datos simbólica).
- P =Número de variables (número de columnas de la tabla de datos simbólicos).
- X =Tabla de datos simbólicos.

Salida :

- Un orden total "$\leq$" sobre el conjunto Ω de objetos.
- Estructura piramidal, es decir, una sucesión de cuadruples $(p, p_I, p_D, f(p))$, con $p = 1, 2, \ldots, NG$, donde NG =número total de gradas de la pirámide, p_I =hijo izquierdo de la grada p y p_D =hijo derecho de la grada p. Si p es una grada terminal entonces $p_I = p_D = 0$.
- Un objeto simbólico O_p asociado a la grada p, con $p = 1, 2, \ldots, NG$.
- La extensión del objeto asociado a cada nodo, es decir, $ext(O_p)$, con $p = 1, 2, \ldots, NG$.
- Si el algoritmo falla la salida será un mensaje de error.

Paso 1: Fase de inicialización

Paso 1.1 $h = 1$, donde h es el número de iteraciones.

Paso 1.2 $NG = N$, donde NG =número total de gradas de la pirámide.

Paso 1.3 $NC = N$, donde NC =Número de componentes conexas, en una iteración dada (al final de la ejecución del algoritmo se tendrá $NC = 1$).

Paso 1.4 $NP = N$, donde NP =Número de gradas activas en una iteración dada (al final de la ejecución del algoritmo se tendrá $NP = 1$).

Paso 1.5 Se inicializan los N primeros cuadruples de la estructura piramidal, como sigue: $(s, 0, 0, 0)$, $s = 1, 2, \ldots, N$.

Paso 1.6 Se construyen NC componentes conexas iniciales $C_s = \{s\}$, $s = 1, 2, \ldots, NC$, y un orden total $\leq_C$ asociado a cada componente conexa, en el que inicialmente se tiene que $s \leq_C s$. Además de denota por C al conjunto formado por todas las componentes, es decir, $C = \{C_1, C_2, \ldots, C_{NC}\}$.

Paso 1.7 Se construyen NP gradas activas iniciales de la forma $G_q = \{(\alpha, \beta, s_q, \ell)\}$, para $q = 1, 2, \ldots, NP$. α es un número que se asocia a cada grada activa en una iteración dada (las gradas activas estarán numeradas de 1 hasta NP), β es el número global de la grada (la primera grada generada por el algoritmo $\beta = N + 1$, para la segunda grada generada por el algoritmo $\beta = N + 2$ y así sucesivamente), s_q es el vector de datos simbólicos almacenado en la fila $q-$ésima de la tabla de datos simbólicos (al inicio cada fila de la matriz corresponde a una grada, sin embargo, cuando el algoritmo avanza una grada puede corresponder a la unión de varios objetos simbólicos, es decir, podrá estar asociada a la "unión de varias filas de la tabla de datos simbólicos) y ℓ es el número de veces que la grada ha sido agregada ($\ell \leq 2$). Se denota por $G = \{G_s\}_{s=1,2,\ldots,NP} = \{(1, 1, s_1, 0), (2, 2, s_2, 0), \ldots, (NP, NP, s_{NP}, 0)\}$ el conjunto de todas las gradas activas iniciales, se denota por $G_q^1 = \alpha$, $G_q^2 = \beta$, $G_q^3 = s_q$ y $G_q^4 = \ell$

Paso 1.8 Se calcula la matriz de disimilitudes inicial $D_{ij}^h = G(s_i \cup s_j)$ donde s_k es el vector de datos simbólicos almacenado en la fila $k-$ésima fila de la tabla de datos simbólica, con $i, j = 1, 2, \ldots, N$.

Paso 2: Fase de eliminación

Paso 2.1 Se encuentra que gradas son agregables y con cuales gradas son agregables usando el criterio de la definición 26, es decir, calcula la matriz:

$$B_{lu} = \begin{cases} 1 & \text{si} \quad G_l \text{ y } G_u \text{ son agregables} \\ 0 & \text{si} \quad G_l \text{ y } G_u \text{ no son agregables} \\ 0 & \text{si} \quad \exists\, \widetilde{G} \in \mathcal{P} \text{ tal que } G_l \text{ es una grada interior } \widetilde{G} \\ 0 & \text{si} \quad \exists\, \widetilde{G} \in \mathcal{P} \text{ tal que } G_u \text{ es una grada interior } \widetilde{G} \end{cases}$$
$$\text{para } l, u = 1, 2, \ldots, NP.$$

Paso 2.2 Calcula las gradas activas que ya no son agregables con ninguna otra grada (o sea ya no serán activas), es decir, encuentra todas las gradas $G\eta$ tal que, la fila y la columna η de la matriz B contienen solamente ceros. Sea $\widetilde{G} = \{G_{\alpha_1}, G_{\alpha_2}, \ldots, G_{\alpha_m}\}$ con $m \geq 0$ el conjunto de las m gradas no más agregables encontradas.

Paso 2.3 $NP = NP - m$.

Paso 2.4 $G = G \setminus \widetilde{G}$.

Paso 2.5 Actualiza la matriz de distancias D^h de modo que:
$D^h \in M_{(NP-m)\times(NP-m)}$, pues se eliminan de D^h todas las filas y columnas asociadas a gradas no activas.

Paso 3: Fase de formación de nuevas gradas (Paso de Generalización)

Paso 3.1 Encuentra s_i y s_j tal que $D_{ij}^h = G(s_i \cup s_j)$ sea mínimo y $B_{ij} = 1$, donde $i, j = 1, 2, \ldots, NP$. Las gradas donde se alcanza este mínimo se denotan por $s_{i^\star}$

y $s_{j^\star}$. Si $B_{ij} = 0$, $\forall\ i, j = 1, 2, \ldots, NP$ entonces el algoritmo termina y retorna un mensaje de error, si no continua en el paso 3.2.

Paso 3.2 $NG = N + h$, seguidamente calcula el siguiente cuadruple de la estructura piramidal $(NG, G^2_{i^\star}, G^2_{j^\star}, D^h_{i^\star j^\star})$.

Paso 3.3 Calcula $s^\star = s_{i^\star} \cup s_{j^\star}$ y su extensión $ext(s^\star)$.

Paso 3.4 Si $s^\star$ es completo y $ext(s^\star) = ext(s_{i^\star}) \cup ext(s_{j^\star})$ entonces el algoritmo continua en el paso 4, si no se toma $B_{i^\star j^\star} = 0$ y el algoritmo regresa al paso 3.1.

Paso 4: Fase de actualización

Paso 4.1 $h = h + 1$.

Paso 4.2 (Actualización de las componentes) Si $G_{i^\star} \in C_{\sigma_1}$ y $G_{j^\star} \in C_{\sigma_2}$ tal que $\sigma_1 \neq \sigma_2$ (pertenecen a componentes conexas diferentes[5])

Paso 4.2.1 Se forma una nueva componente conexa $C_\sigma = C_{\sigma_1} \cup C_{\sigma_2}$, luego en C_σ se define un nuevo orden total, para esto existen cuatro posibilidades:

Caso 1: Si máx$(G_{i^\star})$ = máx(C_{σ_1}) y mín$(G_{j^\star})$ = mín(C_{σ_2}) :

$$\text{Si } x, y \in C_\sigma \text{ entonces } x \leq_{C_\sigma} y \Leftrightarrow \begin{cases} x \leq_{C_{\sigma_1}} y & \text{si} \quad x, y \in C_{\sigma_1} \\ x \leq_{C_{\sigma_2}} y & \text{si} \quad x, y \in C_{\sigma_2} \\ x \in C_{\sigma_1} \quad \text{y} \quad y \in C_{\sigma_2} \end{cases}$$

Caso 2: Si máx$(G_{i^\star})$ = máx(C_{σ_1}) y máx$(G_{j^\star})$ = máx(C_{σ_2})[6]:

$$\text{Si } x, y \in C_\sigma \text{ entonces } x \leq_{C_\sigma} y \Leftrightarrow \begin{cases} x \leq_{C_{\sigma_1}} y & \text{si} \quad x, y \in C_{\sigma_1} \\ y \leq_{C_{\sigma_2}} x & \text{si} \quad x, y \in C_{\sigma_2} \\ x \in C_{\sigma_1} \quad \text{y} \quad y \in C_{\sigma_2} \end{cases}$$

Caso 3: Si mín$(G_{i^\star})$ = mín(C_{σ_1}) y mín$(G_{j^\star})$ = mín(C_{σ_2})[7]:

$$\text{Si } x, y \in C_\sigma \text{ entonces } x \leq_{C_\sigma} y \Leftrightarrow \begin{cases} y \leq_{C_{\sigma_1}} x & \text{si} \quad x, y \in C_{\sigma_1} \\ x \leq_{C_{\sigma_2}} y & \text{si} \quad x, y \in C_{\sigma_2} \\ x \in C_{\sigma_1} \quad \text{y} \quad y \in C_{\sigma_2} \end{cases}$$

Caso 4: Si mín$(G_{i^\star})$ = mín(C_{σ_1}) y máx$(G_{j^\star})$ = máx(C_{σ_2})[8]:

$$\text{Si } x, y \in C_\sigma \text{ entonces } x \leq_{C_\sigma} y \Leftrightarrow \begin{cases} y \leq_{C_{\sigma_1}} x & \text{si} \quad x, y \in C_{\sigma_1} \\ y \leq_{C_{\sigma_2}} x & \text{si} \quad x, y \in C_{\sigma_2} \\ x \in C_{\sigma_1} \quad \text{y} \quad y \in C_{\sigma_2} \end{cases}$$

Paso 4.2.2 $NC = NC - 1$.

Paso 4.2.3 $C = (C \setminus \{C_{\sigma_1}, C_{\sigma_2}\}) \cup \{C_\sigma\}$.

Paso 4.3 (Actualización de las gradas activas)

[5]Si $\sigma_1 = \sigma_2$ evidentemente el algoritmo deja las componentes tal y como estaban.
[6]Notese se realiza una "inversión" de los elementos de C_{α_2}.
[7]Notese se realiza una "inversión" de los elementos de C_{α_1}.
[8]Nótese que se realiza una "inversión" de los elementos de C_{α_1} y de los elementos de C_{α_2}.

Paso 4.3.1 Se calcula la nueva grada[9]: $G_\sigma = G_{i^\star} \cup G_{j^\star} := \{(G^1_{i^\star}, N+h, s_{i^\star} \cup s_{j^\star}, 0)\}$ y se actualiza el número de veces que estas dos gradas han sido agregadas, es decir, $G^4_{i^\star} = G^4_{i^\star} + 1$ y $G^4_{j^\star} = G^4_{j^\star} + 1$. Luego se eliminan (desactivan) las gradas que han sido agregas dos veces, para esto existen cuatro posibilidades:

Caso 1: Si $G^4_{i^\star} = 2$ y $G^4_{j^\star} = 2$ (ambas gradas han sido agregados dos veces) entonces: $NP = NP - 1$ y $G = (G \setminus \{G_{i^\star}, G_{j^\star}\}) \cup \{G_\sigma\}$.

Caso 2: Si $G^4_{i^\star} = 1$ y $G^4_{j^\star} = 1$ (ambas gradas han sido agregados una vez) entonces: $NP = NP + 1$ y $G = G \cup \{G_\sigma\}$.

Caso 3: Si $G^4_{i^\star} = 2$ y $G^4_{j^\star} = 1$ ($G_{i^\star}$ ha sido agregada dos veces y $G_{j^\star}$ ha sido agregada una vez) entonces: $G = (G \setminus \{G_{i^\star}\}) \cup \{G_\sigma\}$.

Caso 4: Si $G^4_{i^\star} = 1$ y $G^4_{j^\star} = 2$ ($G_{j^\star}$ ha sido agregada dos veces y $G_{i^\star}$ ha sido agregada una vez) entonces: $G = (G \setminus \{G_{j^\star}\}) \cup \{G_\sigma\}$ (Solo se calculan las "distancias" que no habian sido calculadas antes).

Paso 4.4 Calcula la nueva matriz de "distancias" $D^h_{ij} = G(s_i \cup s_j)$ para $i, j = 1, 2, \ldots, NP$.

Paso 5: Si $NP = 1$ entonces el algoritmo termina, en caso contrario si $h > M$ entonces el algorimo retorna un mensaje de error, si no regresa al paso 2.

Nota 1 En el Paso 3.1 el mínimo podría alcanzarse en varias parejas de objetos simbólicos. Si se desea encontrar una Pirámide no "Saturada" (ver en [5, Diday (1984)] la definición) entonces se debe escoger la pareja de gradas (objetos simbólicos) de modo tal, que estén lo más lejanas[10] posible en la componente conexa a la que pertenecen (o en la que pertenecerán luego de efectuar la agregación). Si por el contrario se quiere una pirámide con el máximo posible de gradas entonces se debe escoger la pareja de gradas (objetos simbólicos) de modo tal que queden lo más cerca posible en la componente conexa a la que pertenecen (o en la que pertenecerán luego de efectuar la agregación). Actualmente esto es una opción en el programa PYRAMIDE.EXE implementado por los autores para ejecutar este algoritmo.

Nota 2 Cuando la tabla de datos simbólicos tiene datos de tipo histograma, entonces la extensión de los objetos simbólicos se calcula como sigue $ext(e_j) = \{s \text{ tal que } w^s_j \leq w_j\}$ ([4, Brito (1997)]), usando la notación de la definición 6.

El siguiente algoritimo permite construir una pirámide simbólica cuando el orden de los objetos se conoce a priori. Este algoritmo es en realidad un caso particular del algoritmo anterior, ya que se inicia con $n = 1$ componentes conexas, mientras que el algoritmo anterior CAPS inicia con $n = |\Omega|$.

[9] Es importante notar que la inversión en una componente implica la inversión todas las gradas que pertenecen a esta componente, pues el orden de los elementos de las gradas es por definición heredado del orden de los elementos en la componente.

[10] Lo más lejos posible de acuerdo al orden total $\leq_C$ asociado a la componente conexa.

ALGORITMO DE CLASIFICACIÓN ASCENDENTE PIRAMIDAL SIMBÓLICA CON UN ORDEN DADO (CAPSO)

Entrada :

- M =Número máximo de iteraciones.
- N =Número de vectores de datos simbólicos (número de la filas de tabla de datos simbólica).
- P =Número de variables (número de columnas de tabla de la datos simbólicos).
- X =Tabla de datos simbólicos.
- Un orden total "$\leq_\Omega$" sobre el conjunto Ω de objetos.

Salida :

- Estructura piramidal, es decir, una sucesión de cuadruples $(p, p_I, p_D, f(p))$, con $p = 1, 2, \ldots, NG$, donde NG =número total de gradas de la pirámide, p_I =hijo izquierdo de la grada p y p_D =hijo derecho de la grada p. Si p es una grada terminal entonces $p_I = p_D = 0$.
- Un objeto simbólico O_p asociado a la grada p, con $p = 1, 2, \ldots, NG$.
- La extensión del objeto asociado a cada nodo, es decir, $ext(O_p)$, con $p = 1, 2, \ldots, NG$.
- Si el algoritmo falla la salida será un mensaje de error.

Paso 1: Fase de inicialización

Paso 1.1 $h = 1$, donde h es el número de iteraciones.

Paso 1.2 $NG = N$, donde NG =número total de gradas de la pirámide.

Paso 1.3 $NC = 1$, donde NC =Número de componentes conexas, en una iteración dada.

Paso 1.4 $NP = N$, donde NP =Número de gradas activas en una iteración dada (al final de la ejecución del algoritmo se tendrá $NP = 1$).

Paso 1.5 Se inicializan los N primeros cuadruples de la estructura piramidal, como sigue: $(s, 0, 0, 0)$, $s = 1, 2, \ldots, N$.

Paso 1.6 Se construye una componente conexa $C = \{s_1, s_2, \ldots, s_N\}$, con orden total $\leq_C$, definido como sigue: $s_i \leq_C s_j \Leftrightarrow s_i \leq_\Omega s_j$.

Paso 1.7 Se construyen NP gradas activas iniciales de la forma $G_q = \{(\alpha, \beta, s_q, \ell)\}$, para $q = 1, 2, \ldots, NP$. α es un número que se asocia a cada grada activa en una iteración dada (las gradas activas estarán numeradas de 1 hasta NP), β es el número global de la grada (la primera grada generada por el algoritmo $\beta = N + 1$, para la segunda grada generada por el algoritmo $\beta = N + 2$ y

así sucesivamente), s_q es el vector de datos simbólicos almacenado en la fila $q-$ésima de la tabla de datos simbólicos (al inicio cada fila de la matriz corresponde a una grada, sin embargo, cuando el algoritmo avanza una grada puede corresponder a la unión de varios objetos simbólicos, es decir, podrá estar asociada a la "unión de varias filas de la tabla de datos simbólicos) y ℓ es el número de veces que la grada ha sido agregada ($\ell \leq 2$). Se denota por $G = \{G_s\}_{s=1,2,\ldots,NP} = \{(1,1,s_1,0),(2,2,s_2,0),\ldots,(NP,NP,s_{NP},0)\}$ el conjunto de todas las gradas activas iniciales, se denota por $G_q^1 = \alpha$, $G_q^2 = \beta$, $G_q^3 = s_q$ y $G_q^4 = \ell$

Paso 1.8 Se calcula la matriz de disimilitudes inicial $D_{ij}^h = G(s_i \cup s_j)$, donde s_k es el vector de datos simbólicos almacenado en la fila $k-$ésima fila de la tabla de datos simbólica, con $i,j = 1,2,\ldots,N$.

Paso 2: Fase de eliminación

Paso 2.1 Encuentra que gradas son agregables y con cuales gradas son agregables usando el criterio de la definición 26, es decir, calcula la matriz:

$$B_{lu} = \begin{cases} 1 & \text{si} \quad G_l \text{ y } G_u \text{ son agregables} \\ 0 & \text{si} \quad G_l \text{ y } G_u \text{ no son agregables} \\ 0 & \text{si} \quad \exists\, \widetilde{G} \in \mathcal{P} \text{ tal que } G_l \text{ es una grada interior } \widetilde{G} \\ 0 & \text{si} \quad \exists\, \widetilde{G} \in \mathcal{P} \text{ tal que } G_u \text{ es una grada interior } \widetilde{G} \end{cases}$$
$$\text{para } l,u = 1,2,\ldots,NP.$$

Paso 2.2 Calcula las gradas activas que ya no son agregables con ninguna otra grada (o sea ya no serán activas), es decir, encuentra todas las gradas $G\eta$ tal que la fila y la columna η de la matriz B contienen solamente ceros. Sea $\widetilde{G} = \{G_{\alpha_1}, G_{\alpha_2}, \ldots, G_{\alpha_m}\}$ con $m \geq 0$ el conjunto de las m gradas no más agregables encontradas.

Paso 2.3 $NP = NP - m$.

Paso 2.4 $G = G \setminus \widetilde{G}$.

Paso 2.5 Actualiza la matriz de distancias D^h de modo que:
$D^h \in M_{(NP-m)\times(NP-m)}$, pues se eliminan de D^h todas las filas y columnas asociadas a gradas no activas.

Paso 3: Fase de formación de nuevas gradas (Paso de Generalización)

Paso 3.1 Encuentra s_i y s_j tal que $D_{ij}^h = G(s_i \cup s_j)$ sea mínimo y $B_{ij} = 1$, donde $i,j = 1,2,\ldots,NP$. Las gradas donde se alcanza este mínimo se denotan por $s_{i^\star}$ y $s_{j^\star}$. Si $B_{ij} = 0$, $\forall\, i,j = 1,2,\ldots,NP$, entonces el algoritmo termina y retorna un mensaje de error, si no continua en el paso 3.2.

Paso 3.2 $NG = N + h$, luego calcula el siguiente cuadruple de la estructura piramidal $(NG, G_{i^\star}^2, G_{j^\star}^2, D_{i^\star j^\star}^h)$.

Paso 3.3 Calcula $s^\star = s_{i^\star} \cup s_{j^\star}$ y su extensión $ext(s^\star)$.

Paso 3.4 Si $s^\star$ es completo y $ext(s^\star) = ext(s_{i^\star}) \cup ext(s_{j^\star})$ entonces el algoritmo continua en el paso 4, si no se toma $B_{i^\star j^\star} = 0$ y el algoritmo regresa al paso 3.1.

Paso 4: Fase de actualización

Paso 4.1 $h = h + 1$.

Paso 4.2 (Actualización de las gradas activas)

Paso 4.2.1 Se calcula la nueva grada: $G_\sigma = G_{i^\star} \cup G_{j^\star} := \{(G^1_{i^\star}, N + h, s_{i^\star} \cup s_{j^\star}, 0)\}$ y se actualiza el número de veces que estas dos gradas han sido agregadas, es decir, $G^4_{i^\star} = G^4_{i^\star} + 1$ y $G^4_{j^\star} = G^4_{j^\star} + 1$. Luego se eliminan (desactivan) las gradas que han sido agregadas dos veces, para esto existen cuatro posibilidades:

Caso 1: Si $G^4_{i^\star} = 2$ y $G^4_{j^\star} = 2$ (ambas gradas han sido agregados dos veces) entonces: $NP = NP - 1$ y $G = (G \setminus \{G_{i^\star}, G_{j^\star}\}) \cup \{G_\sigma\}$.

Caso 2: Si $G^4_{i^\star} = 1$ y $G^4_{j^\star} = 1$ (ambas gradas han sido agregados una vez) entonces: $NP = NP + 1$ y $G = G \cup \{G_\sigma\}$.

Caso 3: Si $G^4_{i^\star} = 2$ y $G^4_{j^\star} = 1$ ($G_{i^\star}$ ha sido agregada dos veces y $G_{j^\star}$ ha sido agregada una vez) entonces: $G = (G \setminus \{G_{i^\star}\}) \cup \{G_\sigma\}$.

Caso 4: Si $G^4_{i^\star} = 1$ y $G^4_{j^\star} = 2$ ($G_{j^\star}$ ha sido agregada dos veces y $G_{i^\star}$ ha sido agregada una vez) entonces: $G = (G \setminus \{G_{j^\star}\}) \cup \{G_\sigma\}$.

Paso 4.3 Calcula la nueva matriz de "distancias" $D^h_{ij} = G(s_i \cup s_j)$ para $i, j = 1, 2, \ldots, NP$ (Solo se calculan las "distancias" que no habian sido calculadas antes).

Paso 5: Si $NP = 1$ entonces el algoritmo termina, en caso contrario si $h > M$ entonces el algorimo retorna un mensaje de error, si no regresa al paso 2.

2.3. Teoremas de convergencia

Diday en [5, Diday (1984)] propone el siguiente algoritmo (llamado CAP) para construir una pirámide numérica:

El algorimo inicia con la escongencia de un índice de agregación y luego tiene los siguientes etapas:

a) Cada elemento de Ω está en la pirámide y se le llama Grupo.

b) Se agragan los 2 grupos más próximos que no han sido agregados 2 veces.

c) Se regresa al Paso b) hasta que se forme un grupo que contenga Ω.

Además el algoritmo tiene las siguientes condiciones.

d) Cada vez que un grupo se forma se le asocia un orden sobre los 2 grupos que han sido reunidos.

e) Dos grupos no pueden ser reunidos si ellos no son conexos.

f) Sean i y j los elementos extremos de de la parte conexa de Ω asociada a un grupo h; ningún grupo puede ser conectado a un grupo incluido dentro de h que no contiene ni i ni j.

Lema 1 El algorimo CAP construye una pirámide.

Demostración: Se puede consultar en [5, Diday (1984)].

Teorema 1 El algoritmo CAPS construye una pirámide simbólica.

Demostración: La etapa a) del algoritmo CAP la ejecutan los pasos 1.5 y 1.7 del algoritmo CAPS, la etapa b) de CAP la llevan a cabo los pasos 3.1 y 4.3 de CAPS y la etapa c) de CAP es esquivalente al Paso 5 de CAPS.

La condición d) del algoritmo CAP queda establecida en el Paso 4.3.1 del algoritmo CAPS. Mientras que las condiciones e) y f) de CAP las garantizan el Caso 1 y el Caso 2 de la definición 26 respectivamente. Entonces usando el Lema 1 se concluye que la salida de CAPS cumple la condición 1 de la definición 14.

Finalmente el Paso 3.4 del algoritmo CAPS garantiza que la salida será una Pirámide Simbólica, pues en este paso se verifica la completitud del objeto simbólico generado por la nueva agregación. Si este objeto simbólico no es completo, el mínimo es descartado y se regresa al Paso 3.1 hasta encontrar un par de gradas que cumplan las condiciones de agregación y que generen un objeto simbólico completo. Si no existen tales gradas entonces CAPS retornará un mensaje de error. Esto garantiza que si el algoritmo finaliza entonces produce una Pirámide Simbólica , con lo que queda establecida la condición 2 de la definición 14.

Corolario 1 El algoritmo CAPSO construye una pirámide simbólica.

Demostración: Es evidente que el algoritmo CAPSO es un caso particular del algoritmo CAPS.

2.4. Ejemplo

Para ilustrar el funcionamiento del algoritmo en esta sección se presentan dos ejemplos de ejecuciones del algoritmo CAPS.

Ejemplo 2 En este ejemplo se presenta la ejecución del algoritmo CAPS con la siguiente tabla de datos simbólicos. En esta matriz de datos simbólicos se tienen cinco variables, la primera es de tipo intervalo, la segunda es una variable cuantitativa discreta, y las tres últimas variables son de tipo histograma (los valores están truncados).

$$X = \begin{bmatrix} [1,4] & 2 & (1(0{,}4), 2(0{,}1), 3(0{,}2), 4(0{,}07), 5(0{,}02)) & (1(0{,}1), 2(0{,}9)) & (1(0{,}7), 2(0{,}2)) \\ [1,4] & 3 & (1(0{,}6), 2(0{,}1), 3(0{,}1), 5(0{,}0)) & (1(0{,}1), 2(0{,}9)) & (1(0{,}7), 2(0{,}2)) \\ [1,5] & 2 & (1(0{,}7), 2(0{,}2)) & (1(0{,}0), 2(0{,}9)) & (1(0{,}7), 2(0{,}2)) \\ [1,4] & 1 & (1(0{,}7), 2(0{,}0), 3(0{,}1), 4(0{,}0), 5(0{,}0), 6(0{,}0), 7(0{,}0)) & (1(0{,}0), 2(0{,}9)) & (1(0{,}7), 2(0{,}2)) \\ [1,4] & 1 & (1(0{,}4), 3(0{,}4), 4(0{,}0), 5(0{,}0)) & (1(0{,}0), 2(0{,}9)) & (1(0{,}8), 2(0{,}1)) \\ [1,6] & 2 & (2(0{,}4), 3(0{,}1), 4(0{,}3), 5(0{,}0), 6(0{,}0), 7(0{,}0)) & (1(0{,}0), 2(0{,}9)) & (1(0{,}7), 2(0{,}2)) \end{bmatrix}$$

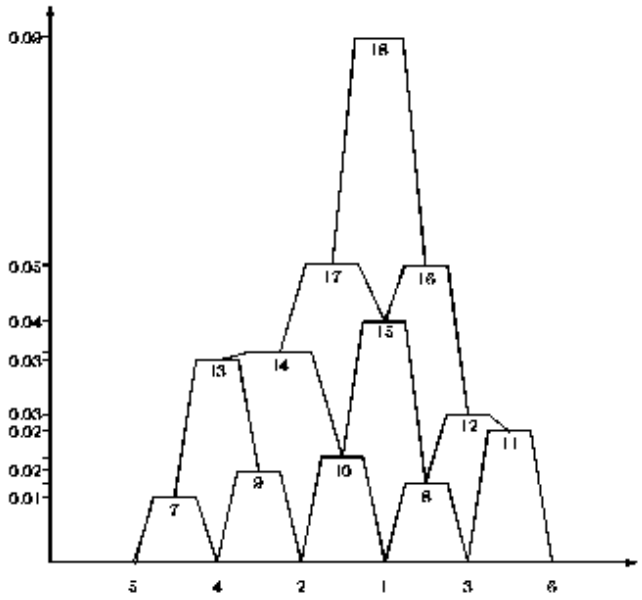

Figura 1: Pirámide de la tabla de datos del ejemplo 2.

La siguiente información corresponde a los objetos simbólicos y sus respectivas extensiones calculadas por el algoritmo CAPS, asociados a cada uno de los nodos de la pirámide.

```
Where the the labels of variables are:
y1=Number of adults older than 16 years
y2=QWEtv-licence
y3=Fuel type central heating
y4=Central heating install
y5=CH repairs last 12 month

Where the labels of the individuals are:
1="Northern metropolitan"
2="North non-metropolitan"
3="Yorks and humberside metropoli"
4="Yorks and humberside non-metro"
5="East midlands non-metropolitan"
```

```
6="Northern ireland"

P7=[y1=[1.000,4.000]]^[y2={1.00}]^[y3=(1(0.7181),2(0.0537),3(0.4348),
   4(0.0870),5(0.0435),6(0.0134),7(0.0067))]^[y4=(1(0.0435),2(0.9799
   ))]^[y5=(1(0.8696),2(0.2483))]
Ext(P7)={4,5}
P8=[y1=[1.000,5.000]]^[y2={2.00}]^[y3=(1(0.7882),2(0.1151),3(0.2806),
   4(0.0791),5(0.0288),6(0.0000),7(0.0000))]^[y4=(1(0.0588),2(0.9856)
   )]^[y5=(1(0.7765),2(0.2734))]
Ext(P8)={1,3}
P9=[y1=[1.000,4.000]]^[y2={3.00,1.00}]^[y3=(1(0.7181),2(0.1259),3(0.
   1879),4(0.0134),5(0.0070),6(0.0134),7(0.0067))]^[y4=(1(0.0201),2(
   0.9860))]^[y5=(1(0.7692),2(0.2483))]
Ext(P9)={2,4}
P10=[y1=[1.000,4.000]]^[y2={2.00,3.00}]^[y3=(1(0.6853),2(0.1259),3(0.
    2806),4(0.0791),5(0.0288),6(0.0000),7(0.0000))]^[y4=(1(0.0144),2(
 0.9860))]^[y5=(1(0.7692),2(0.2734))]
Ext(P10)={1,2}
P11=[y1=[1.000,6.000]]^[y2={2.00}]^[y3=(1(0.7882),2(0.4107),3(0.2000),
    4(0.3750),5(0.0089),6(0.0446),7(0.0179))]^[y4=(1(0.0588),2(0.9643)
    )]^[y5=(1(0.7768),2(0.2235))]
Ext(P11)={3,6}
P12=[y1=[1.000,6.000]]^[y2={2.00}]^[y3=(1(0.7882),2(0.4107),3(0.2806),
    4(0.3750),5(0.0288),6(0.0446),7(0.0179))]^[y4=(1(0.0588),2(0.9856)
    )]^[y5=(1(0.7768),2(0.2734))]
Ext(P12)={1,3,6}
P13=[y1=[1.000,4.000]]^[y2={1.00,3.00}]^[y3=(1(0.7181),2(0.1259),3(0.4
    348),4(0.0870),5(0.0435),6(0.0134),7(0.0067))]^[y4=(1(0.0435),2(0.
    9860))]^[y5=(1(0.8696),2(0.2483))]
Ext(P13)={2,4,5}
P14=[y1=[1.000,4.000]]^[y2={3.00,1.00,2.00}]^[y3=(1(0.7181),2(0.1259),3
    (0.4348),4(0.0870),5(0.0435),6(0.0134),7(0.0067))]^[y4=(1(0.0435),2
    (0.9860))]^[y5=(1(0.8696),2(0.2734))]
Ext(P14)={1,2,4,5}
P15=[y1=[1.000,5.000]]^[y2={2.00,3.00}]^[y3=(1(0.7882),2(0.1259),3(0.28
    06),4(0.0791),5(0.0288),6(0.0000),7(0.0000))]^[y4=(1(0.0588),2(0.98
    60))]^[y5=(1(0.7765),2(0.2734))]
Ext(P15)={1,2,3}
P16=[y1=[1.000,6.000]]^[y2={2.00,3.00}]^[y3=(1(0.7882),2(0.4107),3(0.28
    06),4(0.3750),5(0.0288),6(0.0446),7(0.0179))]^[y4=(1(0.0588),2(0.98
    60))]^[y5=(1(0.7768),2(0.2734))]
Ext(P16)={1,2,3,6}
P17=[y1=[1.000,5.000]]^[y2={2.00,3.00,1.00}]^[y3=(1(0.7882),2(0.1259),3
    (0.4348),4(0.0870),5(0.0435),6(0.0134),7(0.0067))]^[y4=(1(0.0588),2
    (0.9860))]^[y5=(1(0.8696),2(0.2734))]
Ext(P17)={1,2,3,4,5}
P18=[y1=[1.000,6.000]]^[y2={2.00,3.00,1.00}]^[y3=(1(0.7882),2(0.4107),3
    (0.4348),4(0.3750),5(0.0435),6(0.0446),7(0.0179))]^[y4=(1(0.0588),2
    (0.9860))]^[y5=(1(0.8696),2(0.2734))]
Ext(P18)={1,2,3,4,5,6}
```

Cada grada de la pirámide puede ser interpretada, por ejemplo la grada P12 es un nodo de regiones donde "Number of adults older than 16 years" está entre 1 y 6. El número de licencias de televisión (QWEtv-licence) es 2. El "Fuel type central heating" es 1 máximo

en un 78,82 % de los casos, 2 máximo en un 41,07 % de los casos, 3 máximo en un 28,06 % de los casos, 4 máximo en un 37,5 % de los casos, 5 máximo en un 2,88 % de los casos, 6 máximo en un 4,46 % de los casos, 7 máximo en un 1,79 % de los casos. "Central heating install" es 1 máximo en un 5,8 % y 2 máximo en un 98,56 %. "CH repairs last 12 month" es 1 máximo en un 77.68 % de los casos y es 2 máximo en un 27.34 % de los casos.

Referencias

[1] Bertrand, P. (1986) *Etude de la Représentation Pyramidale*. Thèse de 3-ème Cycle, Université Paris IX-Dauphine.

[2] Bertrand, P.; Diday, E. (1990) "Une géneralisation des arbres hiérarchiques: Les représentations pyramidales", *Statistique Appliquée* **38**(3): 53–78.

[3] Brito, P. (1991) *Analyse de Données Symboliques: Pyramides d'Héritage*. Thèse de Doctorat, Université Paris 9 Dauphine.

[4] Brito, P. (1998) "Symbolic clustering of probabilistic data", in: A. Rizzi, M. Vichi & H.H. Bock (Eds.) *Adavances in Data Science and Classification*, Springer-Verlag, Berlin: 385–390.

[5] Diday E. (1984) "Une représentation visuelle des classes empiétantes", Rapport INRIA n. 291. Rocquencourt, France.

[6] Diday E., Lemaire J., Pouget J., Testu F. (1982) *Eléments d'Analyse des Données*. Dunod, Paris.

[7] Diday E. (1987) "Introduction à l'approche symbolique en Analyse des Données", in *Proc. Premières Journées Symbolique-Numérique*, Université Paris IX Dauphine. Décembre 1987.

[8] Diday, E. (1998) "L'Analyse des données symboliques: un cadre théorique et des outils", Cahiers du CEREMADE, Université de Paris IX-Dauphine.

[9] Diday, E.; Bock H.-H. (Eds.) (2000) *Analysis of Symbolic Data. Exploratory Methods for Extracting Statistical Information from Complex Data*. Springer-Verlag, Heidelberg.

[10] Gil, A.; Capdevila, C.; Arcas, A. (1998) "On the efficiency and sensitivity of a pyramidal classification algorithm", Economics working paper 270, Universitat Pompeu Fabra, Barcelona.

[11] Mfoumoune, E. (1998) *Les Aspects Algorithmiques de la Classification Ascendante Pyramidale et Incrémentale*. Thèse de Doctorat, Université Paris 9 Dauphine.

[12] Pollaillon, G. (1998) *Organisation et Interprétation par les Treillis de Galois de Données de Type Multivalué, Intervalle ou Histogramme*. Thése de Doctorat, Université Paris 9 Dauphine.