NUEVOS RESULTADOS EN CLASIFICACION BIMODAL JERARQUICA Y POR PARTICIONES

William Castillo Elizondo[1] y Javier Trejos Zelaya[2],
Centro de Investigación en Matemática Pura y Aplicada, Escuela de Matemática, Universidad de Costa Rica, 2060 San José, Costa Rica

RESUMEN
Se presenta un estudio de los principales métodos de clasificación bimodal, tanto jerárquica como por particiones. Se propone una fórmula de recurrencia del tipo Lance & Williams para la clasificación jerárquica bimodal, usando el criterio de agregación de Eckes & Orlik. También se implementa la técnica de optimización global de sobrecalentamiento simulado para el particionamiento bimodal, obteniéndose mejores resultados que los métodos usuales de intercambios alternantes y de nubes dinámicas o *k*-means.

Palabras clave: clasificación automática; análisis de conglomerados; tablas de contingencia; fórmula de recurrencia; clasificación jerárquica; particionamiento; sobrecalentamiento simulado; optimización global.

ABSTRACT
We present a study of the main methods for two-mode classification, hierarchical and partitioning. A Lance & Williams recurrence formula is proposed for hierarchical two-mode classification using the Eckes & Orlik criterion. Also, it is implemented the simulated annealing technique for global optimization in two-mode partitioning, obtaining better results than usual methods like alternating exchanges and dynamic clusters or *k*-means.

Key words: automatic classification; cluster analysis; contingency tables; recurrence formula; hierarchical classification; partitioning; simulated annealing; global optimization.

1. INTRODUCCION

En clasificación bimodal, se supone dada una matriz de datos no negativos $X = (x_{ij})_{pxq}$ que cruza dos modos I y J con $p = |I|$, $q = |J|$ y $I \cap J = \varnothing$. La matriz X puede ser una tabla de contingencia, una tabla de similitudes u otra, de tal forma que la entrada x_{ij} indica la intensidad de la relación entre la fila i y la columna j. Los métodos de clasificación bimodal buscan particiones del modo fila I y del modo columna J, en forma simultánea, de manera que haya una relación entre las particiones de ambos modos.

Al igual que en los métodos usuales de clasificación automática, también conocidos como de análisis de conglomerados o *clustering*, se pueden distinguir dos tipos principales de métodos: los métodos llamados jerárquicos y los métodos llamados de particionamiento.

A continuación presentamos nuevos resultados obtenidos en clasificación jerárquica bimodal, referentes a una fórmula de recurrencia que permite simplificar los cálculos, y en clasificación bimodal por particiones, relativos al uso de la técnica de sobrecalentamiento simulado con el fin de evitar los mínimos locales que obtienen los métodos usuales para un criterio análogo al de la inercia intra-clases.

2. CLASIFICACION JERARQUICA

2.1 Clasificación jerárquica unimodal

La clasificación jerárquica clásica o unimodal trata de construir árboles jerárquicos que representen clases de objetos de un conjunto $I = \{1,\ldots,n\}$ encajadas unas en otras, que se pueden representar en forma de un árbol tal que las clases unitarias o individuales se representan en las hojas (normalmente dibujadas en la parte baja del árbol), y asociadas a un valor pequeño de un índice, mientras que las clases numerosas se encuentran en las partes superiores del árbol. Puede consultarse Bock (1974), Diday **et al**. (1982) o Hartigan (1975) para una descripción de los principales métodos de clasificación jerárquica.

[1]Fax: +(506) 207 4397; E-Mail: wcastill@cariari.ucr.ac.cr
[2]Fax: +(506) 207 4397; E-Mail: jtrejos@cariari.ucr.ac.cr

Se dispone de una tabla de datos unimodal X de dimensiones $n \times p$, con base en la cual se pueden calcular disimilitudes d_{ij} entre las parejas de filas de X (llamadas los objetos a clasificar). En clasificación jerárquica unimodal, se quiere construir una *jerarquía H* (que llamaremos aquí jerarquía o sistema unimodal) de subconjuntos de I tal que: (i) $I \in H$, $\varnothing \notin H$; (ii) $\forall\, i \in H$: $\{i\} \in H$; y (iii) $\forall h_1, h_2 \in H$: $h_1 \cap h_2 = \varnothing \Rightarrow h_1 \subseteq h_2$ ó $h_2 \subseteq h_1$. La jerarquía se dice *binaria* si $\forall\, B \in H$, $|B| > 1$, $\exists h, h' \in H$: $h \cap h' = \varnothing$ y $h \cup h' = B$. Además, se define un índice $f : H \rightarrow \boldsymbol{R}^+$ tal que f vale cero en los conjuntos unitarios $\{i\}$ y si $h_1 \subseteq h_2$ entonces $f(h_1) \leq f(h_2)$. En este último caso, le jerarquía se dice indexada.

La construcción del árbol se dice *ascendente* cuando se parte de las clases unitarias, y se van formando las clases de los niveles más altos hasta construir la última clase que contiene a todos los objetos. El *algoritmo de clasificación jerárquica ascendente unimodal* (**CJA1**) procede así:

0. Sea $k := 0$; $P_0 := \{\{i\} / i \in I\}$; escoger un criterio de agregación δ
1. Reunir las dos clases h_1, h_2 de P_k más cercanas de acuerdo con δ
2. $k := k + 1$; $P_k = (P_{k-1} \cup \{h_1 \cup h_2\}) - \{h_1, h_2\}$;
3. Ir al paso 1 hasta que $P_k = I$
4. Sea $H := \cup_k P_k$

Debe notarse que el criterio de agregación δ define una disimilitud entre conjuntos de objetos. Además, para aplicar el paso 1 después de la primera iteración, se debe calcular el nuevo valor de δ entre la clase recién formada y las clases que permanecen en P_k. Por ello, la escogencia de un criterio que permita usar los cálculos hechos hasta ese momento ayudaría a hacer más rápido el algoritmo. Así, Lance & Williams (1966-1967) propusieron una fórmula de recurrencia, que permite calcular el valor de δ entre cualquier $h \in P_k$ y la clase $h_1 \cup h_2$ creada por el algoritmo:

$$\delta(h, h_1 \cup h_2) = a_1\, \delta(h, h_1) + a_2\, \delta(h, h_2) + a_3\, \delta(h_1, h_2) + a_4\, |\delta(h, h_1) - \delta(h, h_2)|$$

donde a_1, a_2, a_3 y a_4 son valores que se deben determinar en cada iteración a partir de los cardinales de h, h_1, h_2. La mayoría de los criterios de agregación usuales cumplen esta fórmula de recurrencia (salto mínimo, salto máximo, salto promedio, salto de Ward) y los valores de los coeficientes para cada uno de ellos se pueden consultar en Diday **et al**. (1982) o en Trejos (1998). Es natural definir el índice de la jerarquía por $f(h_1 \cup h_2) = \delta(h_1, h_2)$.

La fórmula de Lance & Williams fue generalizada por Jambu (1978) al caso de jerarquías indexadas, introduciendo los términos relativos a los valores del índice f en h, h_1 y h_2.

Esta fórmula de recurrencia es muy importante porque permite hacer una economía en los cálculos en el algoritmo CJA1, pero también permite caracterizar los criterios de agregación que pueden presentar inversiones, esto es, aquéllos para los que el valor del índice f es inferior en una iteración posterior. Este tipo de fenómeno crearía problemas en la interpretación de un árbol de clasificación. Bataglj (1981) y Diday (1982) han caracterizado los índices de agregación que cumplen la fórmula de recurrencia de Lance & Williams y de Jambu, respectivamente, tales que no presentan inversiones. Por otra parte, también se pueden definir algoritmos más rápidos (bajo ciertas condiciones), como el algoritmo de vecinos recíprocos (Benzécri (1982)) o el algoritmo acelerado de Bruynooghe que tiene la propiedad de reducibilidad (ver Jambu (1978) o Diday **et al**. (1982)).

2.2 Sistema jerárquico bimodal

En clasificación bimodal, se tiene una matriz de similitudes X de dimensiones p x q, con entradas no negativas y que cruza dos modos I y J tales que $I \cap J = \varnothing$. Una clase bimodal será un conjunto no vacío $R \subseteq I \cup J$ de la forma $R = A \cup B$, con $A \subseteq I$ y $B \subseteq J$, siendo A y B también no vacíos. Se define la unión de clases bimodales como $R_1 \cup R_2 = (A_1 \cup A_2) \cup (B_1 \cup B_2)$, donde $A_1 \cap A_2 = \varnothing$ y $B_1 \cap B_2 = \varnothing$.

Un sistema de clasificación jerárquica bimodal es una familia H de subconjuntos de $I \cup J$, tal que:

- $I \cup J \in H$
- $\forall x \in I \cup J$: $\{x\} \in H$
- $\forall\, R_1, R_2 \in H$: $R_1 \cap R_2 \neq \varnothing \Rightarrow R_1 \subseteq R_2$ ó $R_2 \subseteq R_1$.

Para abreviar, lo llamaremos simplemente *sistema bimodal*. Al igual que para sistemas unimodales, un sistema bimodal se dice binario si cada clase no unitaria de H está formada por la unión de dos clases disjuntas del sistema. Además, se dirá indexado si se puede definir un índice f: $H \rightarrow \boldsymbol{R}^+$ que tiene las mismas propiedades que el índice definido para las jerarquías unimodales.

Con base en los conceptos anteriores, la construcción ascendente de un sistema bimodal procedería, a partir de las clases unitarias $\{x\}$ con $x \in I \cup J$, uniendo en cada paso las clases más cercanas en el sentido de algún criterio δ. Así, el algoritmo de *clasificación jerárquica ascendente bimodal* (**CJA2**) procede así:

0. Sea $k := 0$; $P_0 := \{\{x\} : x \in I \cup J\}$; escoger un criterio de agregación δ
1. Reunir las dos clases de P_k más cercanas de acuerdo con δ, excluyendo pares $(\{s\},\{s'\})$ tales que $s,s' \in I$ ó $s,s' \in J$
2. $k := k + 1$; sea $R = A \cup B$ donde (A,B) minimiza δ; sea $P_k = (P_{k-1} \cup \{R\}) - \{A,B\}$; actualizar δ; ir al paso 1 hasta que $P_k = \{I \cup J\}$
3. Sea $H := \cup_k P_k$

Puede verse que la implementación del algoritmo CJA2 requiere de la definición de un criterio de agregación bimodal δ.

2.3 Criterio de Eckes & Orlik

Eckes & Orlik (1993) estudiaron la definición de un criterio de agregación δ en el algoritmo CJA2 que hemos presentado, para las matrices bimodales X aquí consideradas. En un primer momento, estudiaron la *heterogeneidad interna* de una clase bimodal $R = A \cup B$ definida como:

$$HI(R) = \frac{1}{ab}\sum_{i\in A}\sum_{j\in B}(x_{ij} - \mu)^2$$

donde $a = |A|$, $b = |B|$ y $\mu = \max\{x_{ij} : i \in I, j \in J\}$. Es razonable proponer entonces que el criterio de agregación minimice el incremento de HI, dado por $HI(R_1 \cup R_2) - HI(R_1) - H(R_2)$, siendo R_1 y R_2 las dos clases que se unen. Por ello, Eckes & Orlik (1993) propusieron el siguiente criterio de agregación:

$$\delta_{EO}(R_1, R_2) = \frac{1}{a_1 b_2 + a_2 b_1}\left[\sum_{i\in A_1}\sum_{j\in B_2}(x_{ij} - \mu)^2 + \sum_{i\in A_2}\sum_{j\in B_1}(x_{ij} - \mu)^2\right]$$

donde $R_1 = A_1 \cup B_1$, $R_2 = A_2 \cup B_2$, $a_1 = |A_1|$, $b_1 = |B_1|$, $a_2 = |B_2|$, $b_2 = |B_2|$. Puede notarse que esta fórmula incluye los casos particulares:

- $\delta_{EO}(\{i\}, \{j\}) = (x_{ij} - \mu)^2$ para todo $(i,j) \in I \times J$.
- $\delta_{EO}(\{i\}, A \cup B) = \frac{1}{b}\sum_{j\in B}(x_{ij} - \mu)^2, A \times B \subseteq I \times J, B \neq \varnothing$
- $\delta_{EO}(\{j\}, A \cup B) = \frac{1}{a}\sum_{i\in A}(x_{ij} - \mu)^2, A \times B \subseteq I \times J, A \neq \varnothing.$

Por otra parte, aun si $\delta_{EO}(R_1,R_2) \neq HI(R_1 \cup R_2) - HI(R_1) - HI(R_2)$, puede verse como $\delta_{EO}(R_1,R_2)$ es la combinación convexa de heterogeneidades internas:

$$\delta_{EO}(R_1,R_2) = \gamma\, HI(A_1 \cup B_2) + (1 - \gamma)\, HI(A_2 \cup B_1), \text{ con } \gamma = \frac{a_1 b_2}{a_1 b_2 + a_2 b_1}.$$

Es claro que $HI(A_1 \cup B_2)$ y $HI(A_2 \cup B_1)$ son los responsables del incremento de HI cuando se unen R_1 y R_2.

2.4. Fórmula de recurrencia propuesta

Al unir R_1 y R_2 en el algoritmo CJA2, para calcular el criterio de Eckes & Orlik entre $R_1 \cup R_2$ y cualquier otra clase W presente en la partición P_k, se cumple:

$$\delta_{EO}(R_1 \cup R_2, W) = \alpha\, \delta_{EO}(R_1, W) + (1 - \alpha)\, \delta_{EO}(R_2, W), \text{ con } \alpha = \frac{a_1 y + b_1 x}{a_1 y + b_1 x + a_2 y + b_2 x},$$

donde $R_1 = A_1 \cup B_1$, $R_2 = A_2 \cup B_2$, $W = X \cup Y$, y $R_1, R_2, W \in H$, donde $A_1 \cap A_2 = \varnothing = B_1 \cap B_2$. En efecto, si $R_1 \cup R_2 = C_1 \cup C_2$ con $C_1 = A_1 \cup A_2$ y $C_2 = B_1 \cup B_2$, entonces

$$\begin{aligned}
\delta_{EO}(R_1 \cup R_2, W) &= \delta_{EO}(C_1 \cup C_2, X \cup Y) \\
&= \frac{1}{c_1 y + c_2 x}\left[\sum_{i \in C_1}\sum_{j \in Y}(x_{ij} - \mu)^2 + \sum_{i \in X}\sum_{j \in C_2}(x_{ij} - \mu)^2\right] \\
&= \frac{1}{c_1 y + c_2 x}\left[\sum_{i \in A_1}\sum_{j \in Y}(x_{ij} - \mu)^2 + \sum_{i \in A_2}\sum_{j \in Y}(x_{ij} - \mu)^2 + \sum_{i \in X}\sum_{j \in B_1}(x_{ij} - \mu)^2 + \sum_{i \in X}\sum_{j \in B_2}(x_{ij} - \mu)^2\right] \\
&= \frac{1}{c_1 y + c_2 x}\left[(a_1 y + b_1 x)\delta_{EO}(A_1 \cup B_1, X \cup Y) + (a_2 y + b_2 x)\delta_{EO}(A_2 \cup B_2, X \cup Y)\right] \\
&= \alpha\delta_{EO}(R_1, W) + (1 - \alpha)\delta_{EO}(R_2, W).
\end{aligned}$$

2.5. Consecuencias: no inversiones y aceleración del algoritmo

De acuerdo con la fórmula de recurrencia, el criterio δ_{EO} de Eckes & Orlik satisface el axioma de la mediana:

$$\delta_{EO}(R_1 \cup R_2, W) \geq \min\{\delta_{EO}(R_1, W), \delta_{EO}(R_2, W)\},$$

para todo R_1, R_2, W en H, con $R_1 \cap R_2 = \varnothing$. Esto tiene como consecuencia que se puede proponer un algoritmo de cadena (Benzécri (1982)) para la clasificación jerárquica bimodal, basado en la propiedad de los vecinos recíprocos. Ello permitiría hacer importantes economías en los cálculos involucrados en el algoritmo, cuando la tabla de datos es grande.

Defínase el índice f del sistema bimodal como

- $f(\{x\}) := 0$ para todo $x \in I \cup J$
- $f(R) := \delta_{EO}(R_1, R_2)$ para todo $R = R_1 \cup R_2 \in H$, con $R_1, R_2 \in H$.

Así, por el axioma de la mediana $\delta_{EO}(R_1 \cup R_2, W) \geq f(R_1 \cup R_2)$ para todo W perteneciente a la partición $P_{R1 \cup R2}$ creada justo antes de la unión $R_1 \cup R_2$, con excepción de R_1 y R_2. Una consecuencia de esto es que el sistema bimodal H que usa el criterio de agregación δ_{EO} no presenta inversiones, es decir, si se tiene $D_1 \subseteq D_2$ entonces $f(D_1) \leq f(D_2)$. La prueba de esta propiedad se puede hacer siguiendo el mismo esquema que el presentado por Diday **et al**. (1982) para el caso unimodal.

En efecto, sean R, R', $R \cup R' \in H$ tales que R' es creado por el algoritmo CJA2 antes que R.

Por el axioma de la mediana y por el paso 2. del algoritmo, se deduce la desigualdad $\delta_{EO}(R, R') \geq f(R)$. La estrategia para completar la prueba consiste en:

i) Probar que $\delta_{EO}(R, R') \geq \max\{f(R), f(R')\}$.

ii) Para todo $D, D' \in H$, $D \subseteq D' \Rightarrow f(D) \leq f(D')$.

Prueba de i): es suficiente probar que $\delta_{EO}(R, R') \geq f(R')$.

Si R' fue creado en la iteración k de CJA2, sea $R_1 = R_1' \cup R_1''$ la clase creada en la iteración k+1. Por lo tanto R_1' , $R_1'' \in P_{R'}$.

Luego por el paso 2. de CJA2, es claro que $f(R') \leq \delta_{EO}(R_1', R_1'') \leq f(R_1)$. Si $R_1 = R$ la prueba es completa. Si no, entonces se usa el mismo argumento para deducir que existe $R_2 = R_2' \cup R_2''$ con R_1' , $R_1'' \in P_{R1}$ y $f(R_1) \leq f(R_2)$. De esta manera se construyen R_1 R_2 ,…,$R_L = R$, clases de H, tales que $f(R') \leq f(R_1) \leq \ldots \leq f(R_L) = f(R)$.

Prueba de ii): existen $D = D_1 \subset D_2 \subset \ldots \subset D_N = D'$ tales que $D_{l+1} = D'_{l+1} \cup D''_{l+1}$ y $D_l = D'_{l+1}$ ó $D_l = D''_{l+1}$ para $I = 1,\ldots,N - 1$.

Por i) tenemos $f(D_{l+1}) = \delta_{EO}(D'_{l+1}, D''_{l+1}) \geq \max\{f(D'_{l+1}), f(D''_{l+1})\} \geq f(D_i)$ para $I = 1,\ldots,N$-1, es decir $f(D') \geq f(D)$.

Por lo tanto, H no tiene inversiones.

3. CLASIFICACION BIMODAL POR PARTICIONES

El particionamiento bimodal trata de obtener dos particiones $PI = \{A_1,...,A_r\}$ de I y $PJ = \{B_1,...,B_s\}$ de J, tales que se cumpla un criterio de calidad que se especificará más adelante.

Se dirá que $A_k \times B_l$ es una clase bimodal y que $PI \times PJ$ es una partición bimodal de $I \times J$. El *centro de la clase* $A_k \times B_l$ es un número g_{kl} que mide el grado de intensidad en la relación entre las dos clases; por ejemplo,

$$g_{kl} = \frac{1}{\sum_{i \in A_k} \sum_{j \in B_l} p_i q_j} \sum_{i \in A_k} \sum_{j \in B_l} x_{ij} p_i q_j$$

donde p_i,q_j son pesos para la fila i y la columna j, respectivamente. En el caso en que se tengan pesos iguales para todas las filas ($p_i = 1/p$) y columnas ($q_j = 1/q$), entonces el centro de $A_k \times B_l$ tendría la forma

$$g_{kl} = \frac{1}{a_k b_l} \sum_{i \in A_k} \sum_{j \in B_l} x_{ij} p_i q_j$$

donde a_k (resp. b_l) es el cardinal de A_k (resp. B_l).

3.1. Método de intercambios alternantes

Siguiendo una analogía con el modelo de clasificación aditiva de Shepard & Arabie (1979), Gaul & Schader (1996) estudiaron un modelo aditivo bimodal dado por:

$$x_{ij} = \sum_k \sum_l p_{ik} q_{jl} w_{kl} + c + e_{ij} = \hat{x}_{ij} + c + e_{ij}$$

donde p_{ik} y q_{jl} son las funciones indicadoras de las clases asociadas a A_k y B_l, respectivamente, y w_{kl} expresa la intensidad de la relación intermodal acumulada por $A_k \times B_l$.

Dada la matriz de datos positivos $X = (x_{ij})_{pxq}$ y los número de clases r,s, se quiere encontrar las particiones PI, PJ y $W = (w_{kl})$ tales que se minimice el criterio

$$Z = Z(PI, PJ, W) = \sum_i \sum_j (x_{ij} - c - \hat{x}_{ij})^2 .$$

Si se asume que $c = \bar{x} = \frac{1}{pq}\sum_{i,j} x_{ij}$ modela la parte común de *X* entonces la operación de centraje ($x_{ij} \leftarrow x_{ij} - \bar{x}$) produce *c* = 0, asumiendo $\sum_{i,j} e_{ij} = 0$.

Se puede probar (Gaul, Schader (1996)) que

$$Z = \sum_{k=1}^{r}\sum_{l=1}^{s}\sum_{i\in A_k}\sum_{j\in B_l}(x_{ij} - w_{kl})^2$$

y que, para PI y PJ fijos, Z es mínimo cuando w_{kl} es el centro de $A_k \times B_l$:

$$w_{kl} = \frac{1}{a_k b_l}\sum_{i\in A_k}\sum_{j\in B_l} x_{ij} \qquad (2)$$

Gaul & Schader (1996) propusieron el algoritmo de Intercambios Alternantes para encontrar particiones bimodales que minimicen el criterio *Z*. Se denotará $A_k \xrightarrow{i} A_{k'}$ la transferencia del objeto $i \in I$ de la clase A_k a la clase $A_{k'}$, y análogamente $B_l \xrightarrow{j} B_{l'}$ es la transferencia del objeto $j \in J$ de la clase B_l a la clase $B_{l'}$, y ΔZ denotará el cambio en Z al hacer cualquiera de esas dos transferencias. El algoritmo es como sigue:

1. Definir particiones iniciales *PI, PJ* (usualmente al azar); calcular *W* usando (2)
2. Ejecutar los pasos 2.1 y 2.2 hasta que *Z* no decrezca:
   - 2.1 Para $i \in I$, hacer la transferencia $A_k \xrightarrow{i} A_{k'}$; calcular ΔZ.
     Si $\Delta Z < 0$ entonces redefinir *PI* aceptando la transferencia; actualizar *W* usando (2)
   - 2.2 Para $j \in J$, hacer la transferencia $B_l \xrightarrow{j} B_{l'}$; calcular ΔZ.
     Si $\Delta Z < 0$ entonces redefinir *PJ* aceptando la transferencia; actualizar *W* usando (2)

3.2 Método tipo *k*-means o nubes dinámicas

Los métodos clásicos de particionamiento son los basados en el esquema de *k*-means, también conocido como nubes dinámicas. Habiendo fijado el número de clases y a partir de una partición inicial, se hacen iteraciones sobre dos pasos: calcular los núcleos de las clases, asignar los objetos al núcleo de clase más cercano. En general, se muestra que este tipo de algoritmo converge, gracias a un teorema similar al de Huygens (Diday **et al**. (1982)). En el caso bimodal, se puede adaptar ese esquema definiendo los núcleos de las clases a partir de *W* (ver Govaert (1983), Baier, Gaul & Schader (1997)). El algoritmo es como sigue:

1. Definir particiones iniciales *PI, PJ* (usualmente al azar); calcular *W* usando (2)
2. Ejecutar los pasos 2.1, 2.2, 2.3 y 2.4 hasta que *Z* no decrezca:
   - 2.1 Para *W* y *PJ* fijos, definir $P\tilde{I} = \{\tilde{A}_1, \ldots, \tilde{A}_r\}$ por:

$$i \in \tilde{A}_k \Leftrightarrow \sum_{l=1}^{m}\sum_{j\in B_l}(x_{ij} - w_{kl})^2 = \min_{k'}\sum_{l=1}^{m}\sum_{j\in B_l}(x_{ij} - w_{k'l})^2$$

   - 2.2 Para *W* y $P\tilde{I}$ fijos, definir $P\tilde{J} = \{\tilde{B}_1, \ldots, \tilde{B}_m\}$ por:

$$j \in \tilde{B}_l \Leftrightarrow \sum_{k=1}^{r}\sum_{i\in A_k}(x_{ij} - w_{kl})^2 = \min_{l'}\sum_{k=1}^{r}\sum_{i\in A_k}(x_{ij} - w_{kl'})^2$$

   - 2.3 Calcular *W* usando (2)
   - 2.4 Regresar a 2.1 con $PI := P\tilde{I}, PJ := P\tilde{J}$.

Damos a continuación una prueba de la convergencia del algoritmo. Esta prueba no la hemos encontrado en ninguna de las referencias consultadas.

Se trata de probar que si $PI_n \times PJ_n$, W_n son calculados en la n-ésima iteración, entonces $Z_n = Z(PI_n, PJ_n, W_n)$ es decreciente. En efecto, se puede escribir

$$Z_{n-1} = \sum_k \sum_{i \in A_k^{n-1}} [\sum_l \sum_{j \in B_l^{n-1}} (x_{ij} - w_{kl}^{n-1})^2]$$

El algoritmo (paso 2.1) crea $PI_n := \{A_1^n, \ldots, A_r^n\}$ tal que

$$i \in A_k^n \Leftrightarrow \sum_l \sum_{j \in B_l^{n-1}} (x_{ij} - w_{kl}^{n-1})^2 = \min_{k'} \sum_l \sum_{j \in B_l^{n-1}} (x_{ij} - w_{k'l}^{n-1})^2.$$

Para cada $i \in I$, existen índices de clase k,k' tales que $i \in A_k^n \cap A_{k'}^{n-1}$, esto es

$$\sum_l \sum_{j \in B_l^{n-1}} (x_{ij} - w_{kl}^{n-1})^2 \leq \sum_l \sum_{j \in B_l^{n-1}} (x_{ij} - w_{k'l}^{n-1})^2.$$

Entonces cada término de $Z(PI_n, PJ_{n-1}, W_{n-1})$ es menor o igual que algún término de Z_{n-1}, por lo que $Z(PI_n, PJ_{n-1}, W_{n-1}) \leq Z_{n-1}$.

Similarmente, el paso 2.2 del algoritmo implica $Z(PI_n, PJ_n, W_{n-1}) \leq Z(PI_n, PJ_{n-1}, W_{n-1})$. Finalmente, por (2) se tiene $Z(PI_n, PJ_n, W_n) \leq Z(PI_n, PJ_n, W_{n-1})$. Se concluye entonces que $Z_n \leq Z_{n-1}$.

Una consecuencia del decrecimiento de Z_n, es que entonces el algoritmo converge en el siguiente sentido: como Z_n es decreciente y solo existe un número finito de valores distintos de esta sucesión, entonces existe t tal que $Z_t = Z_{t+1} = \ldots$ y luego

- como $Z_n \leq Z(PI_n, PJ_n, W_{n-1}) \leq Z_{n-1}$ y $Z_n = Z_{n-1}$, entonces para $n \geq t$, $w_{kl}^n = w_{kl}^{n+1}$ para todo $(k,l) \in I_{n+1} \times J_{n+1}$;
- para $n \geq t$, si Z es inyectiva entonces $PI_n = PI_{n+1}$, $PJ_n = PJ_{n+1}$.

Hacemos notar que estos resultados también son válidos en el caso en que algunas clases queden vacías por el algoritmo (Castillo (1999)).

3.3 Implementación del sobrecalentamiento simulado

Los dos métodos anteriores, de Intercambios Alternantes y de Nubes Dinámicas, son métodos de descenso de búsqueda local iterativa, que hacen descender el criterio Z en cada iteración, por lo que pueden converger a mínimos locales de Z. En otros problemas de Análisis de Datos y Estadística donde sucede un fenómeno similar, se han aplicado metaheurísticas de optimización combinatoria, como el sobre-calentamiento simulado, la búsqueda tabú y los programas evolutivos, obteniéndose en general buenos resultados. Este ha sido el caso de las rotaciones varimax oblicuas (Trejos (1992)), del particionamiento en clasificación automática o análisis de conglomerados (ver por ejemplo Piza(1998) o Trejos, Murillo, Piza (1998)), del análisis de proximidades o escalamiento multidimensional (en el caso métrico, ver Trejos & Villalobos (1999a), en el caso con restricciones ver González & Trejos (2000)), y de la regresión no lineal (ver Trejos & Villalobos (1999b)).

En el presente trabajo, proponemos la utilización del sobrecalentamiento simulado (Keikpatrick **et al**. (1983)), que es una técnica de optimización global basada en una analogía con el método físico de sobrecalentamiento o *annealing*, en inglés. Consiste en realizar una marcha aleatoria de los estados de un problema de optimización, aceptando un nuevo estado si el mismo mejora el criterio de optimización o bien con una probabilidad controlada por un parámetro externo llamado *parámetro de temperatura* y denotado c_t. En este caso, se acepta el nuevo estado si el tiraje de un número real aleatorio entre 0 y 1 es menor que $\exp(-\Delta Z/c_t)$. Así, a grandes temperaturas corresponden muchas aceptaciones de estados que hacen empeorar el criterio (característica que permite evitar los óptimos locales) y a bajas temperaturas se trata prácticamente de un algoritmo de descenso. Al lector interesado en una descripción completa del sobrecalentamiento simulado se le recomienda consular Aarts & Korst (1989), donde se describe completamente el método.

Una característica importante del sobrecalentamiento simulado, es que -mediante una modelización por Cadenas de Markov homogéneas, aperiódicas e irreducibles- se prueba que converge asintóticamente con probabilidad uno a un estado cuyo valor es óptimo global de la función a optimizar.

Ahora bien, una implementación en tiempo finito del sobrecalentamiento simulado debe tomar en cuenta las condiciones de convergencia asintótica, que son: (i) *reversibilidad*: debe ser posible regresar a un estado anterior, con la misma probabilidad; (ii) *conexidad*: debe ser posible pasar de cualquier estado a otro mediante una cadena finita de estados intermedios tales que la probabilidad de pasar de uno al siguiente sea estrictamente positiva; y (iii) todos los vecinos de un estado tienen la misma probabilidad de generación. Además, esta implementación en tiempo finito debe definir un *programa de enfriamiento*: (i) qué se entiende por temperatura inicial c_0 o alta temperatura (intuitivamente corresponde a un valor tal que prácticamente todos los estados son aceptados, independientemente si mejoran o no el criterio); (ii) qué se entiende por baja temperatura (debe corresponder al hecho de hacer una búsqueda local, por lo que generalmente se toma como cero); cómo se debe descender el parámetro de temperatura (en analogía con el proceso físico de sobrecalentamiento, este descenso debe ser "suave"; y (iv) cuántas iteraciones se harán con un mismo valor de temperatura (este número corresponde a la longitud de la cadena de Markov asociada y será denotado L_c).

Para el particionamiento bimodal, se definirá un estado como una partición bimodal $PI \times PJ$ y un vecindario será una partición obtenida a partir de $PI \times PJ$ mediante transferencias tipo $A_k \xrightarrow{i} A_{k'}$ o tipo $B_l \xrightarrow{j} B_{l'}$. Luego, la partición $PI \times PJ$ tiene $p(r-1) + q(m-1)$ vecinos.

La generación de nuevos estados seguirá el siguiente esquema, que aplica un algoritmo tipo sobre-calentamiento simulado y donde η es un número real entre 0 y 1 tirado al azar con una distribución uniforme:

1. Definir particiones iniciales *PI, PJ* (usualmente al azar); calcular *W* usando (2); estimar c_0
2. Para $t = 0,1,2,...$ hasta que $c_t \approx 0$, sea $c_t = \gamma c_{t-1}$ y hacer L_c veces para cada t:
   Escoger al azar I ó J con probabilidad ½
   2.1 Si se escoge I, entonces:
      escoger al azar *i* con probabilidad uniforme 1/*p* (*i* está en la clase *k*)
      escoger una clase $k' \neq k$ del primer modo, con probabilidad 1/(*r* -1)
      calcular ΔZ
      si $\Delta Z < 0$ o si $\eta \leq \exp(-\Delta Z/c_t)$ entonces
        hacer la transferencia $A_k \xrightarrow{i} A_{k'}$,
        redefinir *PI* aceptando la transferencia y actualizar *W* usando (2)
   2.2 Si se escoge *J*, entonces:
      escoger al azar *j* con probabilidad uniforme 1/*q* (*j* está en la clase l)
      escoger una clase $l' \neq l$ del segundo modo, con probabilidad 1/(*s* -1)
      calcular ΔZ
      si $\Delta Z < 0$ o si $\eta \leq \exp(-\Delta Z/c_t)$ entonces
        hacer la transferencia $B_l \xrightarrow{j} B_{l'}$,
        redefinir *PI* aceptando la transferencia y actualizar *W* usando (2)

Puede verse que con el esquema anterior se cumplen las condiciones de convergencia asintótica del sobrecalentamiento simulado. En efecto, la reversibilidad se verifica puesto que es posible regresar a una partición bimodal previa con la misma probabilidad, tomando la transferencia opuesta (intercambiando *k,k'* ó *l,l'*, según sea el caso). La conexidad también es fácilmente verificada puesto que, mediante un número finito de transferencias, es posible generar cualquier partición bimodal $PI' \times PJ'$ a partir de una partición bimodal dada $PI \times PJ$. Y finalmente, a partir del esquema se ve que los vecinos de una partición $PI \times PJ$ tienen, cada uno, la misma probabilidad $1/2p\,(r-1) + 1/2q\,(m-1)$ de ser generado.

El programa de enfriamiento que se usa es el siguiente:

- c_0 se estima dando una tasa inicial de nuevos estados aceptados χ_0, que permite estimar, mediante una serie de tirajes en blanco del algoritmo, el valor

$$c_0 = \Delta\overline{Z}^{+} / \{\ln m_2 - \ln[(m_1 + m_2)\chi_0 - m_1]\}$$

donde $\Delta\overline{Z}^{+}$ es el incremento promedio del cambio de Z en esos tirajes, siendo m_2 el número de veces que creció Z y m_1 el número de veces que Z disminuyó. Hemos usado en general $\chi_0 = 0.85$.

- El decrecimiento de la temperatura se ha tomado geométrico: $c_t = \gamma c_{t-1}$ con $\gamma = 0.85$.
- El criterio para detener el algoritmo es $c_t \approx 0$ o si después de un número de iteraciones en c_t la partición no cambia.
- La longitud de las cadenas de Markov será una constante L_c que se fijará de acuerdo con las dimensiones de la tabla de datos.

3.4. Resultados comparativos

Se ha aplicado el método descrito en la sección anterior, sobre varias tablas de datos citadas en la literatura. Los resultados presentados aquí son sobre tres tablas de datos usadas por Gaul & Schader (1996).

3.4.1. Tabla coñac

Se trata de una tabla de contingencia de tamaño 10 x 15 que tiene en fila 10 mensajes publicitarios impresos y en columna 5 marcas de coñac. Las entradas indican el número de personas que asoció el mensaje en fila con la característica en columna. Los datos se presentan en la Tabla 1.

Tabla 1. Datos de características de coñac cruzando mensajes impresos.

Mensajes impresos	Características				
	1	2	3	4	5
1	112	104	73	100	77
2	82	80	98	69	95
3	137	118	86	111	82
4	88	86	93	69	98
5	133	108	69	130	86
6	89	92	118	77	114
7	106	96	86	94	84
8	117	111	82	103	93
9	92	88	111	72	109
10	120	110	81	121	87

La Tabla 2 muestra los resultados obtenidos para los tres métodos aplicados: nubes dinámicas (MND), intercambios alternantes (IA) y sobrecalentamiento simulado (SS), para 3 y 4 clases tanto por filas como por columnas.

La tabla contiene varias columnas. La columnas r y s indican, respectivamente, el número de clases por fila y por columna. La columna MV indica el mejor valor de la Varianza Explicada VE dada por:

$$VE = 1 - \frac{\sum_{i,j} (x_{ij} - \hat{x}_{ij})^2}{\sum_{i,j} (x_{ij} - \overline{x})^2}.$$

donde $\overline{x}$ denota la media aritmética de las entradas de la matriz X. Es claro que el mínimo de Z y el máximo de VE se obtienen para la misma partición. La columna % indica el porcentaje de veces que el método

correspondiente encontró el mejor valor MV reportado, en un número # de ejecuciones del programa. Finalmente, las columnas c_0 y L_c indican la temperatura inicial estimada y la longitud de las cadenas de Markov, respectivamente. Es claro que el método de SS debe usarse un número menor de veces que los otros dos, ya que el esfuerzo computacional es mucho mayor.

Tabla 2. Resultados obtenidos con los datos de coñac para el mejor valor (MV) de la Varianza Explicada con *r* clases por fila y *m* clases por columna; % indica el porcentaje de veces que ese valor fue encontrado luego de aplicar el método un número # de veces. Se aplicaron el método de nubes dinámicas (MND), de intercambios alternantes (IA) y de sobrecalentamiento simulado (SS). Las dos últimas columnas dan los parámetros de la temperatura inicial (c_0) y la longitud de las cadenas de Markov (L_c).

		MND			**IA**			**SS**				
r	*s*	MV	%	#	MV	%	#	MV	%	#	c_0	L_c
3	3	0.832	4.2	1000	0.832	60	500	0.832	69	75	3500	600
4	4	0.918	0.5	200	0.918	20	300	0.918	90	120	3000	675

Puede notarse la clara inferioridad que presenta el método de nubes dinámicas, ya que la esperanza de obtener una buena solución es inferior a 5%, mientras que para los otros dos métodos es mucho mayor. Para 3 clases, tanto IA como SS obtienen resultados similares, sin embargo para 4 clases la superioridad de SS es clara. Ahora bien, se debe tomar en cuenta que el SS es un método mucho más lento que los otros dos. En efecto, para el caso de 3 clases MND duró 1 segundo, IA 0.8 segundos y SS 136 segundos, aplicando 40 cadenas de Markov.

Las clases bimodales junto con sus centros, para 3 clases, se dan en la Tabla 3, mientras que para 4 clases se dan en la Tabla 4.

Tabla 3. Clases bimodales y centros para 3 clases, datos de coñac.

	B_1 = {3,5}	B_2 = {1}	B_3 = {2,4}
A_1 = {2,4,6,9}	7.8	- 9	- 17.6
A_2 = {1,7,8}	- 14.2	14.9	4.6
A_3 = {3,5,10}	- 14.9	33.3	19.6

Tabla 4. Clases bimodales y centros para 4 clases, datos de coñac.

	B_1 = {2}	B_2 = {4}	B_3 = {5,3}	B_4 = {1}
A_1 = {2,4}	- 13.7	- 27.7	- 0.7	- 11.7
A_2 = {6,9}	- 6.7	- 22.2	16.3	- 6.2
A_3 = {3,5,10}	15.3	23.9	- 14.9	33.3
A_4 = {7,1,8}	6.9	2.3	- 14.2	14.9

3.4.2 Tabla cigarrillos 1

La tabla de cigarrillos 1 es una tabla de contingencia de dimensiones 12 x 13, que cruza 12 marcas de cigarrillo en fila y 13 características sobresalientes de mensajes publicitarios no impresos, referentes a las 12 marcas. Cada casilla indica el número de veces que las personas interrogadas asociaron correctamente la marca y las características correspondientes. La Tabla 5 muestra los resultados obtenidos, con la misma estructura que la Tabla 2.

Tabla 5. Resultados comparativos para los datos de cigarrillos 1 (ver Tabla 2 para consultar el significado de las abreviaciones utilizadas).

		MND			**IA**			**SS**				
r	*s*	MV	%	#	MV	%	#	MV	%	#	c_0	L_c

3	3	0.656	12.3	1000	0.656	21	500	0.656	100	100	20000	750
4	4	0.756	2	650	0.756	11	300	0.756	54	100	80000	1125

Para estos datos es destacable que el sobrecalentamiento simulado encontró en todas las ocasiones el mejor valor obtenido, para el caso de 3 clases, contrastando con el relativamente pobre desempeño de los otros dos métodos. Sin embargo, de nuevo el costo de aplicación del sobrecalentamiento simulado es, para 3 clases, bastante más elevado que para los demás métodos. En efecto, el SS se aplicó para 40 cadenas de Markov empleando 136 segundos, mientras que nubes dinámicas duró 3 segundos e intercambios alternantes 2 segundos. Las clases y los centros se dan en las Tablas 6 y 7.

Tabla 6. Clases bimodales y centros para 3 clases, datos de cigarrillos 1.

	B_1 = {3,5,6,7,9,11,13}	B_2 = {1,4,8,12}	B_3 = {2,10}
A_1 = {5,9}	- 31.7	117.5	211.9
A_2 = {2,3,4,6,8,11,12}	42.8	55.6	- 41.9
A_3 = {1,7,10}	- 5.1	- 31.6	- 125.1

Tabla 7. Clases bimodales y centros para 4 clases, datos de cigarrillos 1.

	B_1 = {6,3,5,9,7}	B_2 = {10,2}	B_3 = {11,13}	B_4 = {1,12,4,8}
A_1 = {3,6,12,2,8,4}	16.5	- 134.7	- 57.5	- 35.9
A_2 = {9}	1.4	- 92.4	186.5	49.8
A_3 = {1,7,10}	- 15.5	211.9	- 72.2	117.5
A_4 = {11,5}	1.9	- 29.7	35.8	27.7

3.4.3. Tabla cigarrillos 2

La tabla de cigarrillos 2 tiene la misma estructura que la de cigarrillos 1, pero con 16 mensajes publicitarios impresos por fila, dos para cada una de 8 marcas. En columna hay 12 marcas de cigarrillos. La Tabla 8 muestra los resultados, confirmándose la superioridad del sobrecalentamiento simulado, pero de nuevo con un mayor costo en tiempo (para 3 clases, 475 segundos para SS, 2 segundos para IA y 4.5 segundos para MND). Las clases bimodales y los centros de éstas son dados en las Tablas 9 y 10.

Tabla 8. Resultados comparativos para los datos de cigarrillos 2 (ver Tabla 2 para consultar el significado de las abreviaciones utilizadas).

		MND			IA			SS				
r	*s*	MV	%	#	MV	%	#	MV	%	#	c_0	L_c
3	3	0.667	4.2	300	0.667	16	800	0.667	100	100	400	1120
4	4	0.740	0.5	350	0.740	8	200	0.740	62	90	1200	1260

Tabla 9. Clases bimodales y centros para 3 clases, datos de cigarrillos 2.

	B_1 = {2,3,4,5,6,7,8,9,11,12}	B_2 = {1 }	B_3 = {10}
A_1 = {3,9,16}	- 4.4	45.8	0
A_2 = {7,11}	- 4	- 1.7	42.8
A_3 = {1,2,4,5,6,8,10,12,13,14,15}	0	- 3	- 2.8

Tabla 10. Clases bimodales y centros para 4 clases, datos de cigarrillos 2.

	B_1 = {10}	B_2 = {2,3,4,5,6,8,9,11,12}	B_3 = {7}	B_4 = {1}
A_1 = {7,11}	42.8	- 4.8	3.3	- 1.7
A_2 = {1,2,4,5,6,8,10,12,14,15}	- 3	0.1	- 2.7	- 3.1
A_3 = {13}	0	- 3	29.8	- 5.2

A_4 = {3,9,16}	0	- 4.6	- 3.2	45.8

4. CONCLUSIONES Y PERSPECTIVAS

El nuevo algoritmo de particionamiento bimodal usando sobrecalentamiento simulado es claramente superior a los algoritmos de intercambios alternantes y de *k*-means. En las Tablas 2, 5 y 8 se aprecia que el mejor valor obtenido por todos los métodos es el mismo. Sin embargo, se puede notar un mejor rendimiento del método que usa sobrecalentamiento simulado, en el sentido de que al aplicarlo es más probable encontrar la mejor solución que usando cualquiera de los otros dos métodos. Los resultados comparativos presentados muestran que con este nuevo algoritmo la esperanza de obtener una mejor solución es siempre mayor que con los otros algoritmos, e incluso en ocasiones esta esperanza es hasta 10 veces superior.

Por otro lado, la fórmula de recurrencia en clasificación jerárquica bimodal usando el criterio de agregación de Eckes & Orlik, permite implementar de manera eficiente el algoritmo de clasificación jerárquica ascendente y garantiza la no ocurrencia de inversiones.

En un futuro cercano, se tiene pensado extender nuestros trabajos al caso de clasificación trimodal (Eckes & Orlik (1994)), para lo que se necesitará de una fórmula de recurrencia un poco más complicada que la presentada aquí, en caso de que tal fórmula exista. Por su parte, la aplicación del sobrecalentamiento simulado en particionamiento trimodal parece tener una extensión natural. Además, se debe implementar la fórmula de recurrencia para el criterio de Eckes & Orlik, así como el correspondiente algoritmo de vecinos recíprocos.

Finalmente, la aplicación de otras heurísticas de optimización combinatoria, como la búsqueda tabú o los programas evolutivos, puede dar resultados comparables a los obtenidos con sobrecalentamiento simulado.

REFERENCIAS

AARTS, E. and J. KORST (1989): **Simulated Annealing and Boltzmann Machines**, John Wiley & Sons, Chichester.

BAIER, D.; W. GAUL and M. SCHADER (1997): "Two-mode overlapping clustering with applications to simultaneous benefit segmentation and market structuring", in: R. Klar & O. Opitz (Eds.), **Classification and Knowledge Organization**, Springer, Heidelberg: 557-566.

BATAGELJ, V. (1981): "Note on ultrametric hierarchical clustering algorithms", **Psychometrika** 45 (3).

BENZECRI, J. P. (1982): "Contruction d'une classification ascendante hiérarchique par la recherche en chaîne des voisins réciproques", **Les Cahiers de l'Analyse des Données** 7:209-218.

BOCK, H.H. (1974): **Automatische Klassifikation**, Vandenhoecht & Ruprecht, Götingen.

CASTILLO, W. (1999): **Métodos de Clasificación Bimodal y Trimodal**, Tesis de Maestría, Universidad de Costa Rica.

CASTILLO, W. and J. TREJOS (2000): "Recurrence properties in two-mode hierarchical clustering", in: W. Gaul & R. Decker (Eds.), **Classification and Information Processing at the Turn of the Millenium**, Springer, Heidelberg (in press).

DIDAY, E. (1982): "Inversions en classification hiérarchique. Application à la contruction adaptative d'indices d'agrégation", Rapport de Recherche No. 157, INRIA, Le Chesnay.

DIDAY, E.; J. LEMAIRE; J. POUGET and F. TESTU (1982): **Eléments d'Analyse des Données**. Dunod, Paris.

ECKES, T. and P. ORLIK (1993): "An error variance approach for two-mode hierarchical clustering", **Journal of Classification** 10**:**51-74.

_____ (1994): "Three-mode hierarchical cluster analysis of three-way three-mode data", in: H.H. Bock, W. Lenski & M.M. Richter (Eds.), **Information System and Data Analysis. Springer**, Heidelberg: 217-225.

GAUL, W. and M. SCHADER (1996): "A new algorithm for two-mode clustering", in: H.H. Bock & W. Polasek (Eds.), **Data Analysis and Information Systems**. Springer, Heidelberg: 15-23.

GONZALEZ, J. and J. TREJOS (2000): “MDS con restricciones usando sobrecalentamiento simulado”, Preprint CIMPA, Universidad de Costa Rica.

GOVAERT, G. (1983): **Classification Croisée**. Thèse d’Etat, Université de Paris VI-Pierre et Marie Curie.

HARTIGAN, J. (1975): **Clustering Algorithms**. John Wiley & Sons, New York.

JAMBU, M. (1978): **Classification Automatique pour l’Analyse des Données**. Dunod, Paris.

KIRKPATRICK, S; C.D. GELLAT and M.P. VECCHI (1983): “Optimization by simulated annealing”, **Science** 220:671-680.

LANCE, G.N. and W.T. WILLIAMS (1966): “A general theory of classification sorting strategies: 1. Hierarchical systems”, **Computer Journal** 9:373-380.

_____ (1966): “A general theory of classification sorting strategies: 2. Clustering systems”, **Computer Journal** 10**:**271-277.

PIZA, E. (1998): "Clasificación automática: particionamiento mediante sobrecalentamiento simulado", **Investigación Operacional** 19(2-3):152-163.

SHEPARD, R.N. and P. ARABIE (1979): “Additive clustering: representation of similarities as combinations of discrete overlapping properties”, **Psychological Review** 86:87-123.

TREJOS, J. (1982) “A simultated annealing implementation for oblique varimax rotations”, in: J. Janssen & C.H. Skiadas (Eds.), *Applied Stochastic Models and data Analysis*, Vol. II. World Scientific, Singapur: 981-989.

_____ (1998): **Métodos de Clasificación y Discriminación**. Notas de Curso, Maestría en Matemática Aplicada, Universidad de Costa Rica, San José.

TREJOS, J.; A. MURILLO and E. PIZA (1998): “Global stochastic optimization for partitioning”, in: A. Rizzi, M. Vichi & H.H. Bock (Eds**.**), **Advances in Data Science and Classification.** Springer, Heidelberg: 185-190.

TREJOS, J. and M. VILLALOBOS (1999a): “Use of simulated annealing in metric multidimensional scaling”, **International Conference on Large Scale Data Analysis**, Cologne, May 1999.

_____ (1999b) “Optimización mediante recocido simulado en regresión no lineal”, Memorias del **XIII Foro Nacional de Estadística**, AME, Monterrey, México.

TREJOS, J. and W. CASTILLO (2000): “Simulated annealing optimization for two-mode partitioning”, in: W. Gaul & R. Decker (Eds.) **Classification and Information Processing at the Turn of the Millenium**. Springer, Heidelberg (in press).