ITEA (2004), Vol. **100A N.° 1**, 5-16

UN ALGORITMO DE APRENDIZAJE AUTOMÁTICO PARA EVALUACIÓN DE LA CALIDAD A PARTIR DE CALIFICACIONES SUBJETIVAS

Jorge Díez[1,*], Juan José del Coz[**], Oscar Luaces[**], Félix Goyache[2,*], Jaime Alonso[**], Ana María Peña[***], Antonio Bahamonde[**]

* SERIDA-CENSYRA-Somió, c/ Camino de los Claveles 604, E-33203 Gijón (Asturias), España

** Centro de Inteligencia Artificial, Universidad de Oviedo de Gijón, Campus de Viesques, E-33271 Gijón (Asturias), España

*** Facultad de Ingeniería. Universidad Distrital Francisco José de Caldas, Bogotá, Colombia.

RESUMEN

Este trabajo presenta un algoritmo de aprendizaje de funciones para evaluar objetos. Se asume que los conjuntos de entrenamiento del algoritmo provienen de evaluaciones de expertos capaces de ordenar los objetos por su calidad, pero que fallan al evaluar esa calidad mediante un valor absoluto. La situación descrita es típica de la evaluación de la calidad de los productos alimenticios mediante paneles de expertos o consumidores. El algoritmo pretende cubrir la necesidad de la Industria Agroalimentaria de tecnologías eficaces de evaluación de la calidad sensorial para poder proveer a los mercados con productos de calidad uniforme. El algoritmo se basa en una variante de los mapas auto-organizados (SOM) de Kohonen denominada *growing neural gas*, que modifica el tamaño del mapa según las necesidades del problema. El funcionamiento del algoritmo se ha testado sobre problemas artificiales y de acceso público.

Palabras clave: Calidad sensorial, Panel, Redes neuronales, Aprendizaje de preferencias, Inteligencia Artificial.

SUMMARY

A MACHINE LEARNING ALGORITHM TO ASSESS QUALITY FROM SUBJECTIVE CLASSIFICATIONS

In this paper we present an algorithm for learning a function able to assess objects. Our approach assumes that the experts can provide a collection of pair-wise comparisons. However, they have difficulties in assigning a number to the qualities of the objects considered. This is a typical situation when dealing with food products, where it is very interesting to have repeatable, reliable mechanisms that are as objective as possible to evaluate quality in order to provide markets with products of a uniform qua-

1. Dirección actual en Centro de Inteligencia Artificial. Universidad de Oviedo de Gijón, Campus de Viesques, E-33271 Gijón (Asturias), España.

2. Autor al que debe dirigirse la correspondencia.

lity. The algorithm is implemented using a growing variant of Kohonen's Self-Organizing Maps (growing neural gas). Its performance has been tested with a variety of artificial or public data sets to demonstrate the capabilities of our approach.

Key words: Sensorial quality, Panel, Neural networks, Preference learning, Artificial Intelligence.

Introducción

La valoración de la calidad es una tarea compleja. Se pretende condensar todas las características deseables de un objeto en un solo número. Sin embargo, este número no refleja estrictamente un valor absoluto, sino la calidad relativa del objeto respecto de otros. En objetos de origen biológico; la calidad depende de un amplio conjunto de propiedades multisensoriales que son el resultado de su composición química, su estructura física, la interacción de ambas y la manera en que son percibidos por los sentidos humanos (MEILGAARD *et al.*, 1991). Por otra parte, la evaluación de la calidad de los productos alimenticios puede ser diferente cuando se realiza desde el punto de vista de los expertos o desde el de los consumidores. Los consumidores evalúan la calidad de los productos mediante percepciones de atributos simples a través de los sentidos primarios. En muchos casos no existen especificaciones detalladas del grado de calidad y los expertos pueden adoptar un perfil de calidad que excede considerablemente el requerido por el consumidor (BOLLEN *et al.*, 2002). Esto produce que, frecuentemente, las evaluaciones de calidad obtenidas mediante paneles de expertos o paneles de consumidores no sean comparables (BOLLEN *et al.*, 2002). Además la metodología de la evaluación de la calidad mediante paneles hace que los resultados no sean fiables. Experiencias previas con objetos de origen biológico (GOYACHE *et al.*, 2001a; 2001b; DÍEZ *et al.*, 2003) han demostrado que la complejidad de la tarea de valoración hace que la repetibilidad de las calificaciones humanas tienda a ser baja. Por lo tanto, la precisión de los datos de panel es baja a pesar de que los expertos han sido entrenados exhaustivamente y tienen una gran experiencia en la valoración (GOYACHE *et al.*, 2001a; KUSABS *et al.*, 1998). Los expertos y los consumidores son perfectamente capaces de escoger un objeto frente a otro. Sin embargo, suelen errar cuando se les pide que valoren un objeto con un número (MURRAY *et al.*, 2001). Existe un *efecto lote* que perjudica la valoración. En la práctica, los expertos puntúan diferentes objetos con un sentido relativo, comparando los objetos con los restantes del lote. Un objeto rodeado de otros peores, probablemente obtendrá una puntuación más alta que si se presenta rodeado de objetos mejores. Aunque se puede encontrar inaceptable la variabilidad individual en la puntuación absoluta obtenida al valorar la calidad de un objeto, la posición relativa obtenida en el lote es bastante constante.

La situación expuesta supone un grave problema para la industria agroalimentaria ya que le impide abastecer los mercados con productos de calidad normalizada para satisfacer las demandas del consumidor. Además, los operadores de los mercados alimentarios querrían conocer cuales son los principales parámetros físico-químicos que determinan la valoración de la calidad desde

el punto de vista del consumidor, para así mejorar la aceptación de sus productos.

La Inteligencia Artificial es la parte de las Ciencias de la Computación que trata de simular en una computadora comportamientos que en los seres humanos no dudarían en calificar de inteligentes. Uno de los procesos que se identifican con la inteligencia humana es la capacidad de aprendizaje; por esto, las técnicas que buscan que las máquinas aprendan, el aprendizaje automático (AA), es uno de los temas centrales de la IA. Los algoritmos de AA tratan de sintetizar conocimiento a partir de una fuente de información desorganizada. Estos algoritmos producen pues programas capaces de llevar a cabo tareas útiles para nosotros transcribiendo un conocimiento no estructurado pero que resulta relativamente accesible en un conjunto de instrucciones computables y, por tanto, en conocimiento organizado. Desde otro punto de vista se puede decir que el AA trata de clonar el comportamiento para que sea reproducible por una computadora. De forma operativa, el AA permite construir sistemas basados en el conocimiento a partir de indicaciones sobre qué tipo de respuestas serían esperables frente a un conjunto dado de entradas posibles. Por supuesto, se usará AA siempre que se pueda disponer a una batería de muestras de entradas/salidas que formarán los conjuntos de entrenamiento y en el supuesto de que no se conozca otra forma de construir directamente un sistema basado en el conocimiento. Una descripción detallada de los principios y técnicas de AA aplicables en el campo agroalimentario puede encontrarse en (GOYACHE *et al.*, 2001a). La manera sencilla de construir procedimientos automáticos para calcular la valoración de los objetos es hacerse con un conjunto de valoraciones representativas y aplicar un algoritmo de aprendizaje automático de funciones de regresión como Cubist (CUBIST, 2002), M5 (QUINLAN, 1992), M5' (WANG y WITTEN, 1997), SAFE (System to Acquire Functions from Examples) (QUEVEDO y BAHAMONDE, 1999) o BETS (Bets Examples in Training Sets) (DEL COZ *et al.*, 1999).

En este trabajo se presenta una nueva manera de aprender funciones capaces de valorar objetos partiendo de un conjunto de entrenamiento fiable. El conjunto de entrenamiento está formado por pares de descripciones de objetos, dados como atributos continuos, donde el primer objeto se considera peor que el segundo. El objetivo es una función capaz de cuantificar la calidad de los objetos de la manera más coherente que sea posible con las comparaciones o preferencias suministradas como conjunto de entrenamiento. La idea principal es considerar cada ejemplo de entrenamiento como una indicación de la dirección hacia donde se incrementa la calidad. Así, la diferencia vectorial de los objetos comparados se interpreta como una indicación del comportamiento local de la función de valoración. De esta manera, el algoritmo de aprendizaje crea grupos utilizando una versión de los mapas auto-organizados de Kohonen (SOM) (KOHONEN, 1995), que aumenta según las necesidades (FRITZKE, 1995) donde cada vector de referencia contiene una regla de valoración.

El algoritmo

Se ha desarrollado un algoritmo que aprende a clasificar a partir de un conjunto de entrenamiento formado por comparaciones entre objetos denominado LACE (cuadro 1). Para la descripción del funcionamiento de LACE se sigue a DÍEZ *et al.* (2002), añadiendo un tratamiento de las funciones que reduce su número considerablemente.

Descripción general

Se pretende utilizar como fuente de información un conjunto de preferencias o comparaciones entre objetos, que pueden tener como origen evaluaciones de paneles de calidad. La diferencia normalizada entre cada pareja de objetos comparados es, en sí misma, una función de valoración que podría utilizarse para valorar objetos similares a ellos. Sin embargo, este conjunto de reglas de valoración resulta demasiado numeroso y puede contener reglas erróneas, ya que los expertos han podido cometer errores de apreciación y equivocar su preferencia entre dos objetos. Es por ello que la metodología que se propone utiliza un algoritmo de agrupamiento (GNG; FRITZKE, 1995) que agrupa objetos de similares características para obtener una única función de valoración para todos los objetos del grupo. Los grupos se definen por un vector de referencia, de tal modo que un objeto pertenecerá a un determinado grupo si su distancia respecto del vector de referencia de dicho grupo es menor que su distancia respecto del vector de referencia de cualquier otro grupo.

Una vez definidos los grupos de objetos es posible calcular una función para cada grupo y, en un segundo paso, reducir el número de reglas que definen cada función. Para calcular la función de un grupo se calculan las diferencias normalizadas de todas las comparaciones posibles entre los objetos que le pertenecen y partiendo de la media de dichas diferencias como función inicial, ésta trata de optimizarse utilizando un método basado en el algoritmo OC1 (MURTHY *et al.*, 1994). Es posible que algunas de la reglas generadas por las comparaciones entre objetos de grupo no resulten esenciales para definir la función. Utilizando nuevamente el OC1 se comprueba la calidad de la función obtenida mediante la selección iterativa de pares de reglas. Esto permite reducir el número de reglas que definen cada función y mejorar su compresión y utilidad.

Finalmente, el algoritmo lleva a la misma escala la valoración de cada regla mediante el cálculo de un término independiente y un modificador de pendiente para cada función. Esto permite comparar objetos que pertenezcan a grupos diferentes.

Motivación Geométrica

Sean $\mathbf{u}$ y $\mathbf{v}$ vectores que describen las características de dos objetos evaluados por expertos, resultando que $\mathbf{u}$ es peor que $\mathbf{v}$; en símbolos $\mathbf{u} < \mathbf{v}$. Entonces, se busca una función f tal que $f(\mathbf{u}) < f(\mathbf{v})$. Si se asume que f tiene un comportamiento lineal existirá un vector $\mathbf{w}$ tal que

$$f_w(\mathbf{v}-\mathbf{u}) = \mathbf{u}\cdot\mathbf{w} < \mathbf{v}\cdot\mathbf{w} = f_w(\mathbf{v}) \quad (1)$$

donde para dos vectores $\mathbf{z}$ y $\mathbf{t}$, su producto escalar se representa como $\mathbf{z}\cdot\mathbf{t}$.

Desde el punto de vista geométrico (figura 1), la función f_w representa la distancia al hiperplano $\mathbf{u}\cdot\mathbf{w} = 0$; es decir, el hiperplano de vectores perpendiculares a $\mathbf{w}$. Si se busca un $\mathbf{w}$ considerando únicamente vectores normalizados (es decir $\|\mathbf{w}\| = 1$), la diferencia más amplia entre las valoraciones de $f_w(\mathbf{u})$ y $f_w(\mathbf{v})$ se alcanza cuando $\mathbf{w}$ es el vector normalizado en la dirección de $(\mathbf{v}-\mathbf{u})$. De hecho,

$$f_w(\mathbf{v}-\mathbf{u}) = (\mathbf{u}\cdot\mathbf{w})\cdot\mathbf{w} \le \|\mathbf{v}-\mathbf{u}\|\cdot\|\mathbf{w}\| = \|\mathbf{v}-\mathbf{u}\| =$$

$$= \underbrace{\left(\frac{(\mathbf{v}-\mathbf{u})}{\|\mathbf{v}-\mathbf{u}\|}\cdot\frac{(\mathbf{v}-\mathbf{u})}{\|\mathbf{v}-\mathbf{u}\|}\right)}_{1}\cdot\|\mathbf{v}-\mathbf{u}\| =$$

$$= (\mathbf{v}-\mathbf{u})\cdot\frac{(\mathbf{v}-\mathbf{u})}{\|\mathbf{v}-\mathbf{u}\|} = f_{\frac{(\mathbf{v}-\mathbf{u})}{\|\mathbf{v}-\mathbf{u}\|}}(\mathbf{v}-\mathbf{u}) \quad (2)$$

En general, se parte de una familia de comparaciones

$$\{\mathbf{u}_i < \mathbf{v}_i \ \forall i = 1,\ldots,n\} \quad (3)$$

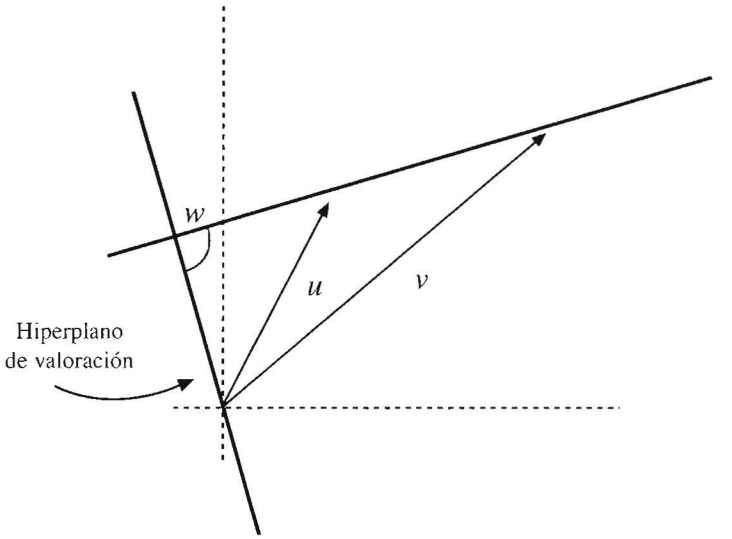


Figura 1. Dados dos objetos representados por los vectores **u** y **v**, si **u** es peor que **v**, el vector normalizado en la dirección de la diferencia, $w = v - u / \|v - u\|$, define un hiperplano. La distancia desde el hiperplano a los objetos es una función de valoración local válida.

*Figure 1. Given tow objects represented by vectors **u** and **v**, if **u** is worse than **v**, the normal vector in direction of the difference, $w = v - u / \|v - u\|$, defines a hyperplane. The distance from the hyperplane to the objects is a suitable local assessment function.*

y se quiere inducir una función f, con un comportamiento lineal en su ámbito local que, se espera, sea capaz de distinguir el mayor número de veces posible que $\mathbf{u}_i$, es peor que $\mathbf{v}_i$, porque $f(\mathbf{u}_i) < f(\mathbf{v}_i)$. El algoritmo que se propone, utiliza la intuición geométrica introducida en esta sección como punto de partida para la búsqueda de la función f. Por tanto, la tarea principal del algoritmo será combinar las pautas locales sugeridas por cada comparación suministrada en el conjunto de entrenamiento.

Agrupamiento de funciones parciales

El conjunto de entrenamiento, compuesto de comparaciones (3), da lugar a un conjunto de $2 \cdot n$ pares de vectores:

$$\left\{\left(\frac{\mathbf{v}_i - \mathbf{u}_i}{\|\mathbf{v}_i - \mathbf{u}_i\|}, \mathbf{u}_i\right) \forall i = 1,\ldots,n\right\} \cup \left\{\left(\frac{\mathbf{v}_i - \mathbf{u}_i}{\|\mathbf{v}_i - \mathbf{u}_i\|}, \mathbf{v}_i\right) \forall i = 1,\ldots,n\right\} \quad (4)$$

La información proporcionada por cada pareja $(\mathbf{w}_i, \mathbf{u}_i)$, debe entenderse como una sugerencia de la regla de calificación numérica (figura 2) que indica que la valoración de un vector **z** es $f_{w_i}(\mathbf{z})$, si **z** se encuentra en las proximidades de $\mathbf{u}_i$, entendiendo que **z** se encuentra en las proximidades de $\mathbf{u}_i$ si la distancia de **z** respecto a $\mathbf{u}_i$ es la menor distancia que mantiene **z** respecto de cualquier otra $\mathbf{u}_j$ contenida en los pares de vectores.

Dado que $f(\mathbf{z}) = \mathbf{z} \cdot \mathbf{w}$, se identificará **w** como la función lineal $f_{\mathbf{w}}$. De la misma mane-

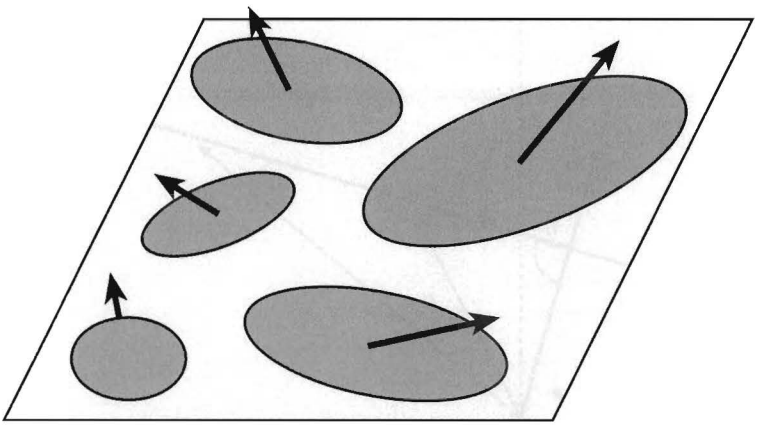


Figura 2. Los grupos de funciones parciales representan, en cada nodo, un entorno en el espacio de atributos de los objetos a ser valorados (coloreados en oscuro en la figura) y un vector apuntando en la dirección en la que se medirán las valoraciones. En otras palabras, el mapa representa un conjunto de reglas de valoración que se aplicarán mediante el criterio de la mínima distancia.

Figure 2. The clusters of partial functions represent, in each node, an environment in the attribute space of the objects to be assessed (drawn dark in the picture) and a vector pointing to the direction to measure the assessments. In other words, the map represents a set of assessment functions to be applied by means of the nearest-distance criterion.

ra, se hará referencia a **u** como las condiciones de la regla. Para abreviar, se escribirá $\mathbf{w} \leftarrow \mathbf{u}$.

En general, se intenta conseguir una función de valoración f definida por partes en todo el espacio de atributos. En otras palabras, la función de valoración vendrá dada por una lista de reglas de calificación numérica

$$(\mathbf{w}_1 \leftarrow \mathbf{u}_1),(\mathbf{w}_2 \leftarrow \mathbf{u}_2),...,(\mathbf{w}_m \leftarrow \mathbf{u}_m) \quad (5)$$

que deben ser evaluadas mediante el criterio de mínima distancia. Matemáticamente, la función f que resulte finalmente inducida trabajará como se muestra para un vector arbitrario **z**

$$f(\mathbf{z}) = \mathbf{w}_k \cdot \mathbf{z} \quad \text{si} \quad \|\mathbf{z} - \mathbf{u}_k\| \leq \|\mathbf{z} - \mathbf{u}\|, \forall j = 1,...,m \quad (6)$$

Una primera tentativa para definir la lista de reglas de valoración (5) es considerar el conjunto de pares al completo (definido en (4)), pero este conjunto de reglas debe mejorarse: es demasiado numeroso y puede contener mucho ruido. Por otro lado, la idea del algoritmo de aprendizaje propuesto (cuadro 1) es unir condiciones similares **u** y después adjuntar una función **w** de acuerdo a las funciones de los pares que han sido unidos en una misma zona (figura 2). Con este fin, se utiliza una versión de los mapas auto-organizados de Kohonen (SOM) (KOHONEN, 1995) que aumenta según las necesidades: *growing neural gas* (GNG) de Fritke (FRITKE, 1995). Este mapa tiene la ventaja de que no necesita que le sean definidos los parámetros a priori como ocurre en el SOM, donde hay que definir la dimensión del mapa y el radio que se va a utilizar para la adaptación.

El algoritmo GNG comienza con un grafo de nodos (vectores de referencia), $\mathbf{c}_1$ y $\mathbf{c}_2$, que representan dos puntos en el dominio de la función de valoración. En cada paso, cada

Cuadro 1. Algoritmo que aprende a clasificar a partir de comparaciones entre objetos (LACE)

Table 1. Algorithm that learn to assess from pair-wise comparison examples (LACE)

Función LACE $(\{\mathbf{u}_i < \mathbf{v}_i : i = 1,...,n\})$: Reglas de valoración

$$E = \left\{\left(\frac{\mathbf{v}_i - \mathbf{u}_i}{\|\mathbf{v}_i - \mathbf{u}_i\|}, \mathbf{u}_i\right) : i = 1,..,n\right\} \cup \left\{\left(\frac{\mathbf{v}_i - \mathbf{u}_i}{\|\mathbf{v}_i - \mathbf{u}_i\|}, \mathbf{v}_i\right) : i = 1,...,n\right\};$$

// para tener valores comparables en [0,1]...
Normalizar cada componente de las condiciones $\mathbf{u}_i$ y $\mathbf{v}_i$ contenidas en E;
// hacer grupos con las condiciones de los ejemplos de E mediante el GNG
GNG (Condiciones(E));
Sean $(\mathbf{w}_1, \mathbf{c}_1)$, $(\mathbf{w}_2, \mathbf{c}_2)$,...., $(\mathbf{w}_m, \mathbf{c}_m)$ los nodos del grafo, donde $\mathbf{w}_i = \frac{1}{|CL_i|}\left(\sum_{j=1}^{|CL_i|} \frac{(\mathbf{v}_j - \mathbf{u}_j)}{\|\mathbf{v}_j - \mathbf{u}_j\|}\right)$
para cada nodo $(\mathbf{w}_i, \mathbf{c}_i)$, del grafo **hacer**

$\mathbf{w}_i = \mathrm{OC1}(CL_i)$

fin para
Unir grupos que puedan compartir una misma función;
Mejorar las pendientes relativas y los términos independientes de las reglas de valoración;
Retorna Reglas de valoración;

iteración, se añade un nuevo nodo tratando de cubrir mejor el espacio. El mapa dejará de crecer y adaptarse cuando alguno de los nodos no clasifique a ningún ejemplo, es decir, cuando exista un nodo que no sea el más cercano a ninguno de los ejemplos que participan en las comparaciones.

Una vez que se tiene un número de grupos representados a través de los vectores de referencia $\mathbf{c}_1, \mathbf{c}_2, \ldots, \mathbf{c}_m$, se considera el conjunto de comparaciones asociadas al vector de referencia $\mathbf{c}_i$, $CL_i = \{\mathbf{u} < \mathbf{v} : (\mathbf{c}_i.\mathbf{u} \le \mathbf{c}_j.\mathbf{u}\ \forall j = 1,K,m) \wedge (\mathbf{c}_i.\mathbf{v} \le \mathbf{c}_j.\mathbf{v}\ \forall j = 1.K,m)\}$, estando compuesto este conjunto por aquellas comparaciones en las que ambos objetos de la comparación tienen como vector de referencia más cercano a $\mathbf{c}_i$. Estas comparaciones se utilizarán para calcular una aproximación lineal de la función de valoración en los alrededores de $\mathbf{c}_i$.

El procedimiento seguido para encontrar una función lineal para el grupo $\mathbf{c}_i$ con coeficientes $\mathbf{w}_i = (a_1,...., a_d)$ se toma del algoritmo OC1 (MURTHY *et al.*, 1994) modificándose ligeramente para este propósito. Lo que se busca es un vector $\mathbf{w}_i$ tal que haga cierto que $(\mathbf{v} - \mathbf{u}) > 0$ para el mayor número de comparaciones posible de CL_i. Se puede empezar con un $\mathbf{w}_i$ que sea la media de las diferencias normalizadas de las comparaciones contenidas en CL_i.

$$\mathbf{w}_i = (a_1,...,a_d) = \frac{1}{|CL_i|}\left(\sum_{j=1}^{|CL_i|} \frac{(\mathbf{v}_j - \mathbf{u}_j)}{\|\mathbf{v}_j - \mathbf{u}_j\|}\right) \quad (7)$$

Ahora, se tratan de mejorar los coeficientes a_m de uno en uno. Una observación importante es que dada una diferencia normalizada $\frac{(\mathbf{v} - \mathbf{u})}{\|\mathbf{v} - \mathbf{u}\|} = (x_1,..., x_d)$, se considera que la comparación que ha dado lugar a esa diferencia está correctamente clasificada, es decir, la función calculada otorga mayor valoración a $\mathbf{v}$ que a $\mathbf{u}$, si el producto escalar de la diferencia por la función es mayor que 0:

$$\mathbf{w}\cdot(\mathbf{v}-\mathbf{u})=\sum_{i=1}^{d}(a_i x_i)>0 \quad (8)$$

De manera equivalente, dicha comparación se clasifica correctamente si:

$$a_m > \frac{-(a_1 x_1 + a_2 x_2 + \ldots + a_{m-1} x_{m-1} + a_{m+1} x_{m+1} + \ldots + a_d x_d)}{x_m} = U \quad (9)$$

o por el contrario $a_m < U$ cuando $x_m < 0$. Cuando $x_m = 0$, el valor del coeficiente a_m no importa. Así, para fijar los valores de todos los coeficientes, cada ecuación (9) representa una restricción sobre el valor de a_m. Partiendo de todos los valores de U ordenados, se considera como valor posible para a_m, el valor que satisface el mayor número de restricciones. Siguiendo el procedimiento de OC1, se repite el ajuste de coeficientes mientras se produzca una mejora.

De esta manera, para cada grupo tenemos una función de valoración local válida en su ámbito de aplicación. Sin embargo, es posible que algunos de esos grupos puedan compartir una misma función. Si se toman los dos grupos más cercanos entre sí y se calcula la función de valoración que mejor se ajusta de manera local a las diferencias normalizadas de ambos grupos, se puede estimar si la capacidad de otorgar una valoración correcta aumenta al tener una función para cada uno de los dos grupos o si con una única función para ambos es suficiente. Repitiendo este proceso a todos los grupos por proximidad, se logra reducir de forma notable el número de funciones necesarias.

En cualquier caso, las reglas de valoración encontradas necesitan un proceso de mejora final. Se parte de que $\mathbf{w} \leftarrow \mathbf{c}$, que por la forma en que ha sido calculada, clasifica correctamente las comparaciones en las que ambos objetos comparten como vector de referencia más cercano a **c**. Pero **w** pude errar cuando uno de los dos objetos va a ser clasificado por otra regla, $\mathbf{w}' \leftarrow \mathbf{c}'$ Para solventar estas situaciones se buscan *modificadores de pendientes* (*a*) y *términos independientes* (*b*) adecuados, de tal manera que la función de la regla de valoración se verá modificada resultando finalmente una regla del tipo:

$$a(\mathbf{w}\cdot\mathbf{z})+b \leftarrow \mathbf{c} \quad (10)$$

donde **z** es un objeto a valorar.

El procedimiento para encontrar *a* y *b* para cada regla de valoración es casi el mismo que se ha descrito para ajustar los coeficientes de a_m de cada **w**. La única diferencia es que ahora se consideran comparaciones donde sólo uno de los objetos se encuentra cerca de la regla que va a ser mejorada.

Resultados experimentales

Con el fin de comprobar la validez del algoritmo se han realizado varios experimentos utilizando problemas artificiales o bien problemas públicamente disponibles (BLAKE y MERZ, 1998). Con el primer grupo de experimentos se pretende ver el comportamiento del algoritmo con problemas de los que se conocen a priori los resultados deseables.

Un problema simple es considerar objetos descritos por un único atributo continuo x con valores en el rango [0,1], y teniendo como función de valoración *verdadera* la parábola (figura 3a). Para construir el conjunto de entrenamiento de comparaciones E, se han generado 3000 pares (x_1, x_2) con valores en el rango [0,1], y se ha añadido el par (x_1, x_2) en E cuando $g(x_1)>g(x_2)$. En el otro caso se ha añadido a E el par (x_2, x_1). El algo-

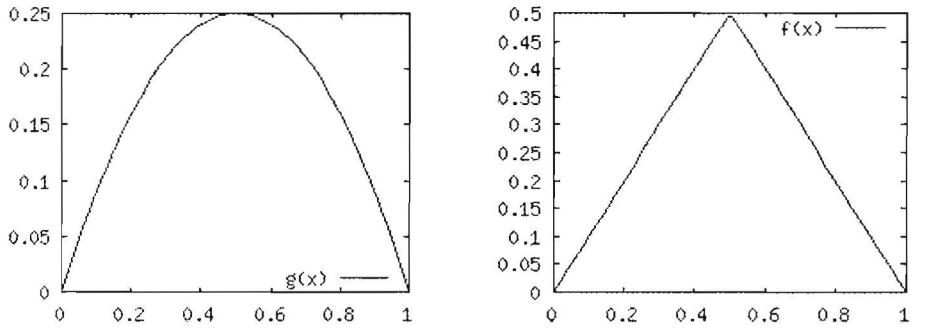


Figura 3. Los objetos a clasificar se encuentran en el rango [0,1]. (a) La clasificación correcta se describe mediante la función g(x) = -*x*(*x*-1). (b) Función aprendida por el algoritmo. Todas las comparaciones fueron ordenadas correctamente por la función (b).

Figure 3. The objects to classify are in the range [0,1]. (a) The correct classification is describe by the function g(x) = -x(x-1). (b) The learned function by the algorithm. All comparisons were correctly ordered by the function (b).

ritmo aprendió para el conjunto de entrenamiento E la función f dibujada en la figura 3b. Aunque los valores de f (*x*) y g(*x*) son bastante diferentes, los valores relativos son prácticamente los mismos. De hecho, probándolo contra un conjunto de comparaciones de test (que ha sido construido de la misma manera que E), todos los pares fueron correctamente ordenados por la función f.

Se llevó a cabo un segundo bloque de experimentos (cuadro 2) con objetos descritos por dos atributos continuos: *x e y*. Se generaron aleatoriamente pares de objetos que fueron pasados al conjunto de comparaciones de entrenamiento colocando primero el objeto del par que se considera peor de acuerdo con la correspondiente función de valoración. De la misma manera (cambiando la semilla de generación de números aleatorios) se generaron también conjuntos de test. Los conjuntos generados contienen 3000 pares. El error que se muestra en la tabla 2 se corresponde con el porcentaje de pares mal ordenados por la función de valoración aprendida por el algoritmo. Se aprecia un porcentaje de error bastante pequeño. El problema en el que una parte del domino se valora con la suma y la otra parte con la resta es un problema más complicado, ya que las funciones son planos ortogonales.

Finalmente, se utilizaron cuatro problemas públicamente disponibles de categoría continua en un intento por tratar con problemas reales. Los experimentos realizados fueron validaciones cruzadas, con 10 particiones y 5 repeticiones, lo que da lugar a 50 pruebas por experimento. En cada prueba, por cada ejemplo del conjunto de entrenamiento, se seleccionaron aleatoriamente otros 10 ejemplos contra los que compararlo: así resultaron 10 comparaciones por cada ejemplo del conjunto. Esta misma operación se realizó en el conjunto de test con el fin de extraer un conjunto de comparaciones con las que validar lo aprendido. Para comparar los resultados de *LACE*, se utilizaron dos conocidos sistemas de aprendizaje de funciones: M5' (QUINLAN, 1992; WANG y

Cuadro 2. Los experimentos llevados a cabo se definieron con funciones compuestas por dos subfunciones con dominios separados. En los dos primeros casos los objetos utilizados en el conjunto de entrenamiento fueron vectores contenidos en [0,999] × [0,999]. En los últimos dos casos los vectores están contenidos en [200,299] × [0,999] y en [700,799] × [0,999]. El conjunto de entrenamiento y el de test contienen 3000 comparaciones

Table 2. Experiments carried out with goal functions defined by two linear subfunctions with separate domains. The original objects to be assessed were vectors in the rectangles [0, 999] × [0,999] in the first two rows, and for the other two [200,299] × [0,999], and [700,799] × [0,999]. Both training and test sets have 3000 elements

Función objetivo	Porcentaje de error
$f(x,y)=\begin{cases} x+10y & x\le 500 \\ 10x+y & x>500 \end{cases}$	2,23
$f(x,y)=\begin{cases} x+y & x\le 500 \\ x-y & x>500 \end{cases}$	8,00
$f(x,y)=\begin{cases} x+10y & x\in[200,299] \\ 10x+y & x\in[700,799] \end{cases}$	0,47
$f(x,y)=\begin{cases} x+y & x\in[200,299] \\ x-y & x\in[700,799] \end{cases}$	1,00

WITTEN, 1997) y Cubist (CUBIST, 2002). Se entrenó con M5' y con Cubist utilizando todo el problema, esto es, considerando la descripción de los ejemplos y la categoría continua asociada a cada uno de ellos. Para evaluar lo que estos sistemas aprendieron, se compararon las valoraciones propuestas para cada objeto de las comparaciones, comprobando así si el orden propuesto era correcto. Los resultados pueden verse en el cuadro 3.

Discusión

Los resultados obtenidos tanto en el aprendizaje de problemas artificiales como en el caso de problemas reales públicamente disponibles demuestran un buen comportamiento de Lace. Los problemas artificiales (cuadro 2) diseñados deberían, *a priori*, suponer un problema fácil de resolver por nuestro algoritmo y así sucede. Los errores obtenidos son bajos, especialmente cuando las regiones con diferente función de valoración están separadas. En la prueba comparativa que se muestra en el cuadro 3, el algoritmo LACE se encuentra penalizado respecto de los otros algoritmos utilizados. La razón es que tanto M5' como Cubist tienen acceso a la valoración continua de todos los ejemplos de entrenamiento, mientras que *LACE* solamente tiene acceso a pares en los que se sabe que hay diferencias, pero sin conocer la cuantía de dichas diferencias. Como ya se ha comentado en la introducción, en algunos problemas reales no se conoce la valoración numérica de los objetos y, por tanto, no se podrían utilizar los algoritmos que aprenden funciones como M5' o CUBIST. A pesar de ello, LACE obtiene resultados similares a los ofrecidos por CUBIST y M5'.

TESAURO (1989), aborda un problema similar al mostrado en el presente trabajo. El algo-

Cuadro 3. Porcentajes de error de *LACE* frente a problemas públicamente disponibles y frente al problema de la parábola descrito anteriormente. CPU y Body Fat fueron obtenidos de la página web de CUBIST, (2002); Boston Housing y Liver Disorders pueden encontrarse en el almacén de la UCI (BLAKE y MERZ, 1998). La comparativa no es justa para LACE porque CUBIST y M5' tienen acceso a las valoraciones otorgadas por los expertos

Table 3. Error scores of our learner in publicly available regression datasets in addition to the parabola dataset described above. The CPU, Body fat were dowloaded from CUBIST URL, while Boston housing, and Liver disorders can be found at UCI Repositoty (BLAKE y MERZ, 1998). The comparative is not fair for LACE *because CUBIST and M5' have access to the assessments carry out by the experts*

Problema	CUBIST	M5'	LACE
CPU	14,27	14,28	14,61
Boston Housing	12,05	11,96	13,42
Body Fat	18,04	18,69	18,92
Liver Disorders	39,11	38,23	41,78
Media	20,87	20,79	22,18

ritmo propuesto por TESAURO (1989) busca funciones capaces de seleccionar la alternativa preferible en su famoso jugador artificial de *Backgammon*. Propone construir una red neuronal simétrica formada por dos subredes separadas, una para cada objeto de la comparación. Además, fuerza a ambas subredes a tener los mismos pesos (solo multiplicada por -1 en la capa de salida). Sin embargo, esta restricción en el mecanismo de entrenamiento únicamente es válida en redes de tipo perceptrón, por lo menos en su campo de aplicación. Otros estudios basados en el perceptrón se describen en UTGOFF y CLOUSE (1991) y en UTGOFF y SAXENA (1987).

Conclusiones

En este documento, se presenta un nuevo método de obtener funciones de valoración de objetos. Este método permite hacer uso de un tipo de conocimiento capaz de ordenar satisfactoriamente un conjunto de objetos del mejor al peor, pero que falla en el cálculo del grado de calidad de un objeto con un número absoluto. Esta aproximación es aplicable sobre clasificaciones realizadas de una manera absoluta que se ven fuertemente afectadas por un *efecto lote*. Ese efecto lote hace que la puntuación obtenida en la clasificación no signifique estrictamente un valor sino un orden de preferencia de unos objetos de un lote respecto de los otros. Nuestra metodología es claramente aplicable a objetos de tipo biológico y especialmente en la industria agroalimentaria (GOYACHE *et al.*, 2001a), en que las reglas para decidir el grado de la calidad de un producto normalmente no están bien definidas, pero el orden de los productos, en cuanto a calidad, es bastante constante y aceptado por parte de los consumidores y expertos.

Bibliografía

BLAKE C.L., MERZ, C.J., 1998. UCI repository of machine learning databases. [http://www.ics.uci.edu/~mlearn/MLRepository.html]. University of California, Irvine, Dept. of Information and Computer Sciences.

BOLLEN A.F., KUSABS N.J., HOLMES G., HALL M.A., 2002. Comparison of consumer and producer perceptions of mushroom quality. En S.E. Prussia W.J. Florkowski y R.L. Shewfelt, editores, *Integrated View of Fruit and Vegetable Quality International Multidisciplinary Conference*, Georgia. 303-311.

CUBIST, 2002. Release 1.09, [http://www.rulequest.com/cubist-info.html].

DEL COZ J.J., LUACES O., QUEVEDO J.R., ALONSO J., RANILLA J., BAHAMONDE A., 1999. Self-organizing cases to find paradigms. *Lecture notes in computer sciences*. 1606, 527-536.

DÍEZ J., DEL COZ J.J., LUACES O., GOYACHE F., PEÑA A.M., BAHAMONDE A., 2002. Learning to assess from pair-wise comparisons. VIII Conferencia Iberoamericana de Inteligencia Artificial (IBERMIA). *Lecture Notes in Artificial Intelligence*. 2527, 481-490. Springer-Verlag.

DÍEZ J., BAHAMONDE A., ALONSO J., LÓPEZ S., DEL COZ J.J., QUEVEDO J.R., RANILLA J., LUACES O., ÁLVAREZ I., ROYO L.J., GOYACHE F., 2003. Artificial Intelligence techniques point out differences in classification performance between light and standard bovine carcasses. *Meat Science*, 63:249-258.

FRITZKE B., 1995. A growing neural gas network learns topologies. En G. TESAURO, D.S. TOURETZKY y T.K. LEEN, editores, *Advances in Neural Information Processing Systems* 7, páginas. MIT Press, Cambridge MA. 625-632.

GOYACHE F., BAHAMONDE A., ALONSO J., LÓPEZ S., DEL COZ J.J., QUEVEDO J.R., RANILLA J., LUACES O., ÁLVAREZ I., ROYO L.J., DÍEZ J., 2001a. The usefulness of artificial intelligence techniques to assess subjetive quality of products in the food industry. *Trends in Food Science and Technology*, 12:370-381.

GOYACHE F., DEL COZ J.J., QUEVEDO J.R., LÓPEZ S., ALONSO J., RANILLA J., LUACES O., ÁLVAREZ I., BAHAMONDE A., 2001b. Using artificial intelligence to design and implement a morphological assessment system in beef cattle. *Animal Science*. 73, 49-60.

KOHONEN T, 1995. Self-organizing maps. *Springer Series of Information Science*.

KUSABS N., BOLLEN F., TRIGG L., HOLMES G., INGLIS S., 1998. Objective measurement of mushroom quality. En *New Zealand Institute of Agricultural Science and the New Zealand Society for Horticultural Science Annual Convention*. New Zealand, 51.

QUEVEDO J.R., BAHAMONDE A., 1999. Aprendizaje de funciones usando inducción sobre clasificaciones discretas. En *CAEPIA' 99-TTIA'99 VIII Conferencia de la Asociación Española para la Inteligencia Artificial – III Jornadas de Transferencia Tecnológica de Inteligencia Artificial*, páginas 64-71.

QUINLAN J.R., 1992. Learning with continuous clases. En *Procedings fifth Australian Joint Conference on Artificial Intelligence*. World Scientific, Singapore. 343-348.

MEILGAARD D., CIVILLE G.V., CARR B.T., 1991. Sensory evaluation techniques. *Boca Raton CRC Press*.

MURRAY J.M., DELAHUNTY C.M., BAXTER I.A., 2001. Descriptive Sensory Analysis: Past, Present and Future. *Food Research International*. 34:461-471

MURTHY S.K., KASIF S., SALZBERG S., 1994. A system for induction of oblique decision trees. *Journal of Artificial Intelligence Research*. 2, 1-32.

TESAURO G., 1989. Connectionist learning of expert preferences by comparison training. *Advances in Neural Information Processing*. 99-106.

UTGOFF P.E., CLOUSE J.A., 1991. Two kinds of training information for evaluation function learning. En *Ninth National Conference on Artificial Intelligence*, páginas 596-600.

UTGOFF P.E., SAXENA S., 1987. Learning a preference predicate. En *Fourth International Workshop on Machine Learning*. 115-121.

WANG Y., WITTEN I.H., 1997. Induction of model trees for predicting continuous classes. En *European Conference on Machine Learning*. 128-137.

(Aceptado para publicación el 22 de enero de 2004).