



Reordenación de los ejemplos en  
Fuzzy-ARTMAP para ganar robustez  
y disminuir el número de categorías:  
aplicación a la extracción automática  
de reglas

A. PEDRO TEJEDOR B. JORGE  
MARTÍNEZ C. EUGENIO F. SÁNCHEZ

Abstract

Contents

Home Page



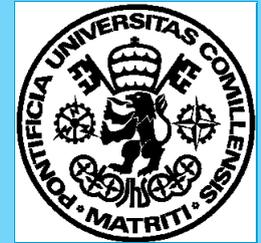
Go Back

Close

Quit

**Keywords:** neural networks, fuzzy-ARTMAP, incremental learning, linguistic rule extraction

**ABSTRACT:** *One of the drawbacks for the fuzzy-ARTMAP algorithm is its dependence on the order of the training samples. When we want to extract linguistic rules from samples, we need a system able to produce as few rules as possible, while maintaining its liability. We propose that an overcome of the difficulty can arise from a recursive reordering of the training samples according with parameters suggested by the algorithm itself.*



A. Pedro Tejedor B. Jorge  
Martínez C. Eugenio F.  
Sánchez

Title Page

Contents



Go Back

Close

Quit

Page 2 of 24

# Contents

|          |   |           |
|----------|---|-----------|
| <b>1</b> | <b>INTRODUCCIÓN</b>                                       | <b>6</b>  |
| <b>2</b> | <b>DESCRIPCIÓN DE FUZZY-ARTMAP</b>                        | <b>8</b>  |
| <b>3</b> | <b>DEBILIDADES DEL ALGORITMO Y PROPUESTAS DE SOLUCIÓN</b> | <b>11</b> |
| <b>4</b> | <b>REORDENACIÓN DE LOS EJEMPLOS DE ENTRENAMIENTO</b>      | <b>14</b> |
| <b>5</b> | <b>EJEMPLO DE APLICACIÓN</b>                              | <b>18</b> |
| <b>6</b> | <b>CONCLUSIÓN</b>   | <b>22</b> |



A. Pedro Tejedor B. Jorge  
Martínez C. Eugenio F.  
Sánchez

Title Page

Contents



Go Back

Close

Quit

Page 3 of 24

# Índice de Figuras

|   |   |    |
|---|---|----|
| 1 | Dinámica de aprendizaje de fuzzy-ARTMAP . . . . .                                   | 10 |
| 2 | Criterios para orden de presentación de los ejemplos dentro de las clases . . . . . | 16 |
| 3 | Evolución del algoritmo . . . . .   | 20 |



A. Pedro Tejedor B. Jorge  
Martínez C. Eugenio F.  
Sánchez

Title Page

Contents



Go Back

Close

Quit

Page 4 of 24

# Índice de Tablas

- 1 Resultados de la clasificación. Medias y desviaciones típicas. . . 19



A. Pedro Tejedor B. Jorge  
Martínez C. Eugenio F.  
Sánchez

Title Page

Contents



Go Back

Close

Quit

Page 5 of 24

# 1. INTRODUCCIÓN

El presente trabajo surge del estudio de la posibilidad de extracción de reglas de las redes Fuzzy-ARTMAP. El problema de la extracción de reglas lingüísticas desde los datos constituye siempre un compromiso entre la inteligibilidad de las reglas generadas y la precisión del modelo. Frecuentemente los modelos más precisos son los más complejos. Por ello es necesario encontrar metodologías capaces de generar conjuntos fiables, robustos y abarcables de reglas. Uno de ellos puede ser Fuzzy-ARTMAP. Sin embargo existen algunos problemas todavía, siendo el más prominente de estos su dependencia del orden de presentación de los ejemplos de entrenamiento.

Fuzzy ARTMAP es un algoritmo de clasificación supervisada capaz de tratar con entradas analógicas y que utiliza la teoría de la resonancia adaptativa (*Adaptive Resonance Theory*). El algoritmo Fuzzy-ARTMAP surge como evolución de los anteriores, ART1, ART2 y ARTMAP [1]. Ha suscitado considerable atención por parte de diversos autores que han profundizado en sus características [5] o propuesto mejoras de tipo estadístico [6], de programación distribuida [4], de cambio de función base [8] o incluso para extraer reglas lingüísticas [3]. Ha demostrado ser capaz de enfrentarse a muy distintos problemas de clasificación, por ejemplo en [7] se ha utilizado para clasificar una base de datos muy dispersa de números manuscritos.

La ordenación de este trabajo sera la siguiente: en primer lugar, describiremos la estructura y dinámica de Fuzzy-ARTMAP. A continuación expondremos algunas de sus dificultades, y las propuestas de solución de los distintos autores, que si bien consiguen resultados de ajuste y precisión, alejan al algoritmo de



A. Pedro Tejedor B. Jorge  
Martínez C. Eugenio F.  
Sánchez

Title Page

Contents



Go Back

Close

Quit

Page 6 of 24

las propiedades deseables para la extracción de reglas, o bien le hacen perder robustez. Propondremos un camino de solución que consiste precisamente en ayudarse del algoritmo para reorganizar los datos de entrada. Finalmente, con un sencillo caso práctico se probarán las soluciones propuestas.



A. Pedro Tejedor B. Jorge  
Martínez C. Eugenio F.  
Sánchez

Title Page

Contents



Go Back

Close

Quit

Page 7 of 24

## 2. DESCRIPCIÓN DE FUZZY-ARTMAP

En su caso más general, fuzzy-ARTMAP está compuesto por dos módulos fuzzy-ART que se ocupan de hacer una agrupación de las entradas en categorías. De ellos uno se encarga de procesar la entrada suministrada al sistema, y el otro la salida. Los llamaremos F-ART-E y F-ART-S. En los casos en que las clases de salida son conocidas, F-ART-S se reduce a mantener un mapa de las clases posibles. Existe un tercer módulo que se encarga de comparar la salida de la categoría elegida con la salida deseada. Es capaz de rechazar la categoría propuesta por el módulo F-ART-E en caso de que no coincidan. Describiremos a continuación someramente el proceso que se sigue a la presentación de un nuevo dato, una entrada con su salida asociada, durante el entrenamiento [2].

Los datos se normalizan previamente. Al vector de entradas se le une el de su complemento a uno, para formar un vector doble de tamaño. Esto tiene una serie de ventajas prácticas, entre las que se encuentran la no proliferación de categorías. Además, nos permitirá usar el operador lógico  $\wedge$  para las operaciones que efectuarán los distintos módulos, como se explicará más adelante. La entrada complementada se presenta al módulo F-ART-E.

El algoritmo de agrupamiento del módulo F-ART-E es el que se encarga en primer lugar de elegir cuál es la categoría más cercana a la entrada presentada. Después evalúa si dicha categoría se tiene un valor de ajuste lo suficientemente grande como para superar un umbral elegido, llamado “factor de vigilancia”. En ese caso considerará que la entrada pertenece a esa categoría. En caso de que no lo supere, elige la siguiente categoría más cercana, y repite la evaluación. Si ninguna de las características existentes cumple con este “criterio de ajuste”,



A. Pedro Tejedor B. Jorge  
Martínez C. Eugenio F.  
Sánchez

Title Page

Contents



Go Back

Close

Quit

Page 8 of 24

se crea una nueva categoría que incluye esta muestra. Si encuentra alguna que cumple con el criterio, presenta su salida al módulo de enlace. Este se encargará de decidir si la salida propuesta coincide con la deseada. En caso de que sea así, se producirán el aprendizaje de la nueva categoría. En caso contrario, se lanza la búsqueda de una nueva categoría.

Introduciremos las ecuaciones principales que gobiernan nuestro proceso mediante un ejemplo sencillo. Sea un espacio de entrada bidimensional. Este presenta la ventaja de que las categorías son representables como rectángulos dados por sus valores extremos y las entradas por puntos. Supongamos que sólo tenemos una categoría creada, de valores  $x = [0.2, 0.4]$ ,  $y = [0.3, 0.7]$ . La entrada presentada será  $[0.7, 0.2]$ . Están representados en la figura 1 por círculos. Codificados con el complemento a 1, estos serán  $\mathbf{w} = [x_{min}, y_{min}, (1 - x_{max}), (1 - y_{max})] = [0.2, 0.3, 0.6, 0.3]$ ,  $\mathbf{I} = [0.7, 0.2, 0.3, 0.8]$ .

Las ecuaciones que nos harán elegir la categoría adecuada o crear una nueva son la de elección de  $T$  máxima (1) y la de comprobación del umbral de resonancia (2)

donde “ $\alpha$ ” es el parámetro de elección. Cuanto mayor sea  $\alpha$  mayor será el número de nodos que se creen, e influirá en el orden de elección de categorías (cf. a este respecto el excelente estudio en [5]). “ $\wedge$ ” es un operador borroso AND, en este caso el mínimo. El significado del módulo es  $|\mathbf{x}| = \sum_i x_i$ . “ $\rho$ ” es el umbral de ajuste, llamado “parámetro de vigilancia”. La ecuación (1) nos permite ordenar las categorías, y elegir la  $T_{max}$ . La segunda define un umbral de nuestra elección que determinará si la categoría está lo suficientemente ajustada a nuestra entrada. En nuestro caso ejemplo los valores de los segundos miembros de las ecuaciones serán ( $\alpha = 0.1$ ): 0.67 para (1) y 0.5 para (2). Si  $\rho < 0.5$  se escogerá esa característica (resonará). En caso contrario se descartará, y se



A. Pedro Tejedor B. Jorge  
Martínez C. Eugenio F.  
Sánchez

Title Page

Contents



Go Back

Close

Quit

Page 9 of 24

## Ecuaciones de la dinámica de Fuzzy-ART

$$T_j = \frac{|\mathbf{I} \wedge \mathbf{w}_j|}{\alpha + |\mathbf{w}_j|} \quad (1)$$

$$\rho < \frac{|\mathbf{I} \wedge \mathbf{w}_j|}{|\mathbf{I}|} \quad (2)$$

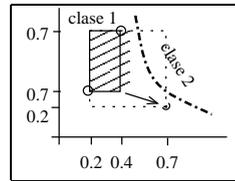
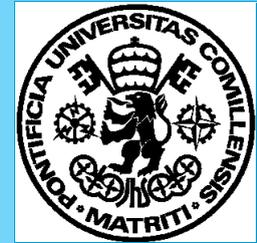


Figura 1: Dinámica de aprendizaje de fuzzy-ARTMAP

buscará otra. Como en nuestro hipotético ejemplo no existe otra, se crearía una nueva categoría, con  $\mathbf{w}_2 = \mathbf{I}$ .

En caso de que resuene, se presenta la salida propuesta por la característica al módulo de enlace. Este está encargado de confrontar esta salida con la salida real que corresponde a nuestro ejemplo. Si ambas coinciden, se inicia el proceso de aprendizaje, que, en su versión más conservadora, sigue el algoritmo  $|\mathbf{w}_{\text{nuevo}}| = |\mathbf{w}_{\text{antiguo}} \wedge \mathbf{I}|$ , tal como está expresado en la figura 1. A medida que aumenta el tamaño de un nodo, disminuyen sus valores. Esto garantiza la convergencia del algoritmo, y permite crear nuevos nodos para “matizar” las divisiones creadas por nodos que han crecido invadiendo otras clases, como está expresado en la figura 1 por el crecimiento de la nueva clase, en línea punteada, para asumir la entrada presentada, traspasando con su esquina superior derecha la frontera de la clase 2, marcada por línea a trazos y puntos. Cuanto menor sean un nodo, más altos serán los valores de sus extremos y será preferido en la elección de categoría [5].



A. Pedro Tejedor B. Jorge  
Martínez C. Eugenio F.  
Sánchez

Title Page

Contents



Go Back

Close

Quit

Page 10 of 24

### 3. DEBILIDADES DEL ALGORITMO Y PROPUESTAS DE SOLUCIÓN

Debido a sus características incrementales, se detectaron desde muy pronto algunas debilidades del algoritmo. Estas provienen del módulo de agrupamiento, Fuzzy-ART, que es el que genera las categorías. En concreto se han apreciado, su falta de robustez frente al ruido y su dependencia del orden de presentación de los ejemplos de entrenamiento.

#### 3.1. Adaptación lenta, múltiples épocas, votación

Estas debilidades han sido atacadas ya desde el primer trabajo sobre Fuzzy-ARTMAP [2]. Se han propuesto tres técnicas: adaptación lenta de los nodos, presentación en múltiples épocas de los mismos datos, entrenamiento simultáneo de varias redes, con votación posterior para la elección de la salida.

La adaptación lenta de los nodos es una técnica comúnmente empleada en los algoritmos de clasificación para ganar robustez cuando hay que clasificar datos que puedan tener un cierto componente de ruido en sus valores. También la presentación de varias épocas de datos se utiliza a menudo. El efecto producido sobre nuestro algoritmo es el de un crecimiento de las categorías y una generación de categorías nuevas, que consigue un mayor ajuste. Finalmente, el entrenamiento de múltiples redes nos permite tener distintas configuraciones de redes, y previsiblemente aumentar el acierto. Esto permite aumentar el grado de confianza que se puede depositar en el algoritmo como un correcto clasificador. Sin embargo, el precio que hay que pagar por estos ajustes es el de un aumento



A. Pedro Tejedor B. Jorge  
Martínez C. Eugenio F.  
Sánchez

Title Page

Contents



Go Back

Close

Quit

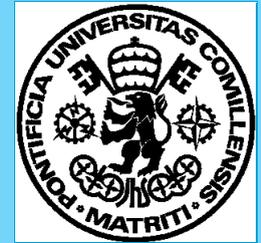
Page 11 of 24

del tamaño del modelo a utilizar. En el caso de las numerosas épocas, aumentará el número de nodos debido a la reclasificación de algunos de los ejemplos que se estaban clasificando mal. En el caso de las múltiples redes, podremos suponer que el número de nodos se multiplica por el número de redes para obtener una estimación del aumento en número de los mismos.

Este aumento en el número de las categorías, aunque mejora la capacidad de previsión de nuestro sistema, lo hace más oscuro a la hora de extraer reglas. Además es fácil comprobar que a menudo se crean nodos espúreos, que nunca se borrarán. De ahí que parezca natural buscar algún método que nos permita eliminar algunos nodos inútiles o perjudiciales.

### 3.2. Poda

Debido a las dificultades aducidas en el párrafo anterior, los creadores del método propusieron una modificación que permitiera eliminar, o “podar” aquellos nodos que, bien por su falta de uso, bien por su poco acierto, bien por ambos motivos, no alcanzaran un cierto grado de confianza [3]. Esto permite disminuir, a veces de forma dramática, el número de nodos que posee nuestro modelo. Sin embargo, si eliminamos los nodos de poco uso, y no permitimos que haya varias redes al mismo tiempo para establecer una estrategia de votación, volveremos a caer en la dependencia del orden de presentación de los ejemplos. Es fácil comprobar la amplia diversidad de modelos que se puede obtener con un mismo conjunto de ejemplos y distintas presentaciones aleatorias. Y su grado de acierto no es una función del número de nodos, ya que no es raro conseguir un mayor acierto con menor número de nodos.



A. Pedro Tejedor B. Jorge  
Martínez C. Eugenio F.  
Sánchez

Title Page

Contents



Go Back

Close

Quit

Page 12 of 24

Será muy deseable, por tanto, encontrar algún mecanismo que nos permita superar la dependencia del orden de presentación de los ejemplos. Esto aumentará su robustez y fiabilidad. Además, previsiblemente disminuirá el número de nodos creados, ya que tenderán a desaparecer los nodos espúreos, fruto de un orden de presentación desafortunado de algunos de los ejemplos.



A. Pedro Tejedor B. Jorge  
Martínez C. Eugenio F.  
Sánchez

Title Page

Contents



Go Back

Close

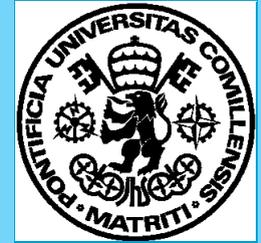
Quit

Page 13 of 24

## 4. REORDENACIÓN DE LOS EJEMPLOS DE ENTRENAMIENTO

Un orden correcto de presentación de los ejemplos de entrenamiento induciría unas categorías óptimas en nuestro modelo. Esto ya se ha propuesto en [2] en el ejemplo de la espiral. En él se presentan los datos que configuran la espiral de fuera a dentro, recorriendo los distintos brazos. Se trata en este caso de aprovechar el conocimiento que ya poseemos sobre el conjunto a clasificar. Sin embargo, esto no siempre es posible, bien porque tal conocimiento no exista, bien porque no esté disponible de forma explícita. Es en esos casos en los que nos queremos ayudar de algún procedimiento que nos permita extraer un número estable de reglas, lo menor posible, que describa ajustadamente nuestro espacio muestral.

Dado que no suponemos ningún conocimiento previo de los datos, tal reordenación sólo puede venir del mismo algoritmo. De esta forma, aprovecharemos el conocimiento que hemos adquirido sobre los datos para mejorar el orden en que se presentan. El algoritmo, los recibirá sin ningún nodo comprometido. Es decir, no es una segunda “época”, porque no reaprovechamos los nodos ya creados. Es una nueva red que estamos entrenando en un orden sugerido por ella misma. Dejamos al algoritmo volver a crear las categorías que estime mejores, permitiendo así variabilidad que no nos estanque en una configuración fruto del azar de la primera presentación.



A. Pedro Tejedor B. Jorge  
Martínez C. Eugenio F.  
Sánchez

Title Page

Contents



Go Back

Close

Quit

Page 14 of 24

## 4.1. Primer criterio

Como lo que buscamos es disminuir y estabilizar el número de categorías, aun a riesgo de perder algo de ajuste, vamos a comenzar presentando aquellos ejemplos que ya han sido asimilados correctamente por alguno de los nodos. A continuación los que están fuera de cualquier nodo. Finalmente, los que han sido mal clasificados. Para ello contamos con el parámetro  $T_{max}$ , que alcanza su mayor valor cuando el ejemplo cae dentro del nodo, y tanto mayor cuanto mayor sea la categoría a la que pertenezca. Esto es debido a la influencia de  $\alpha$  (cf. ecuación (1)).

Sin embargo, esto no será suficiente. Precisamente porque buscamos categorías grandes puede ocurrir que presentemos ejemplos cercanos a cada una de las esquinas de una categoría, creando así nuevos nodos donde antes sólo había uno. Trataremos de evitar esta contingencia suministrando un crecimiento *gradual* de las categorías. Por ello necesitaremos un segundo criterio que nos permita guiar el crecimiento de las clases.

## 4.2. Segundo criterio

La mejor forma de hacer crecer un conjunto convexo de forma gradual es comenzar por el centro y crecer de forma radial, como se puede apreciar en la clase central de la figura 2(a). En ella se han representado las curvas de nivel correspondientes a la distancia euclídea al centro de la clase. En línea punteada con punta de flecha se representa el orden de presentación de los ejemplos de entrenamiento si nos basamos en este criterio. Efectivamente, las categorías aisladas son hipercubos, que constituyen conjuntos convexos. Pero Fuzzy-ARTMAP



A. Pedro Tejedor B. Jorge  
Martínez C. Eugenio F.  
Sánchez

Title Page

Contents



Go Back

Close

Quit

Page 15 of 24



A. Pedro Tejedor B. Jorge  
Martínez C. Eugenio F.  
Sánchez

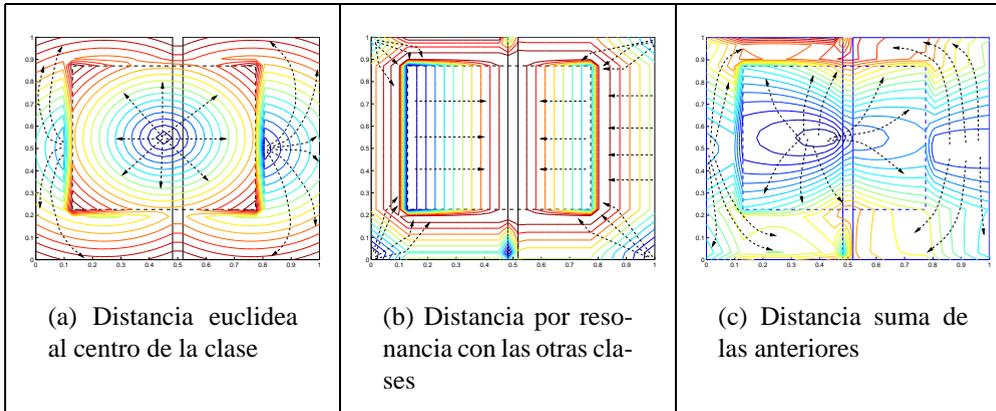


Figura 2: Criterios para orden de presentación de los ejemplos dentro de las clases

Title Page

Contents



Go Back

Close

Quit

Page 16 of 24

frecuentemente genera categorías que intersectan unas con otras, perdiendo así su convexidad. Más aún, a menudo el centro de una categoría está dentro de otra, probablemente de clase de salida distinta. Esto es lo que ocurre en las dos categorías extremas de la figura 2(a), que corresponden a una clase distinta de la clase central. En ese caso, si nos fijamos sólo en la distancia al centro, estaremos suministrando en primer lugar aquellos ejemplos que están en la frontera. Estos generalmente no son los más adecuados para crear categorías amplias y representativas de la clase considerada. Más bien necesitaríamos elegir en primer lugar aquellos ejemplos que están más alejados de las otras categorías. Necesitaremos alguna medida de lo lejos que estamos de las otras categorías, para poder suministrar esos ejemplos en primer lugar.

Esta medida la podemos encontrar de nuevo en el mismo algoritmo. En este caso se ha elegido la suma de los valores deresonancia con el resto de las clases. Esto genera unas líneas de contorno que se pueden apreciar en la figura 2(b). En ella también se han dibujado con trazo discontinuo el orden de presentación de los ejemplos. Esta distancia tiene también problemas: los extremos de nuestra clase alejados por igual de las otras son indiferentes para esta ordenación. Además, las clases convexas pueden empezar su presentación por extremos opuestos.

Probablemente un óptimo para este problema se encuentre en una suma ponderada de ambas distancias, tal como se expresa en la figura 2(c). En ella se han representado las curvas de nivel generadas por la suma ponderada:  $2(c) = 0.6 \cdot 2(a) + 0.4 \cdot 2(b)$ . En ella vemos que el crecimiento previsible de las clases es más suave y en la dirección indicada para todos los casos.



A. Pedro Tejedor B. Jorge  
Martínez C. Eugenio F.  
Sánchez

Title Page

Contents



Go Back

Close

Quit

Page 17 of 24

## 5. EJEMPLO DE APLICACIÓN

A modo de ensayo, hemos aplicado todo lo anterior a un ejemplo sencillo de clasificación. Se ha escogido la forma de un cuadrado en el interior de un cuadrado de dimensión  $1 \times 1$ . El cuadrado interior tiene un área aproximadamente igual a la mitad del área total. Aparece en las tres figuras 2. Se ha elegido este ejemplo por la facilidad que presenta para evaluar los resultados obtenidos. Además, debido a la geometría del problema y a las características de Fuzzy-ARTMAP, se espera que los resultados sean especialmente buenos por su eficiencia. Para todos los ejemplos se ha fijado un mismo conjunto de parámetros: El parámetro de elección ha sido escogido muy conservador,  $\alpha = 0.001$ , porque queremos que cree un número pequeños de nodos. Por idéntica razón se ha escogido  $\rho = 0$ . Como tasa de aprendizaje se ha escogido  $\beta = 0.5$ .

Dado que la reordenación no interfiere con la dinámica del algoritmo, es de esperar que funcione tanto habiendo aplicado la poda como sin aplicarla. Por ello hemos evaluado el método de las dos maneras. Los resultados se muestran en la tabla 1. En ella se han incluido también los resultados referentes al siguiente apartado.

**Sin poda:** Se nota un descenso en el número de nodos, y en su dispersión. Esto de por sí es una característica deseable. Sin embargo, ha aumentado el número de categorías en cuatro de las ejecuciones, y el tanto por ciento de acierto con respecto al conjunto de test ha bajado en 1.64%.



A. Pedro Tejedor B. Jorge  
Martínez C. Eugenio F.  
Sánchez

Title Page

Contents



Go Back

Close

Quit

Page 18 of 24

**Con poda:** Obtenemos unos resultados análogos al caso anterior. Ha bajado el número de nodos y la dispersión en su cantidad. También se han dado tres casos de aumento en el número de nodos, y el tanto por ciento de acierto ha disminuido como media en un 2%.

|          | sin reordenación |          | una reordenación |          |           |          | 50 vueltas (cf. 5.1) |          |           |          |              |          |
|----------|------------------|----------|------------------|----------|-----------|----------|----------------------|----------|-----------|----------|--------------|----------|
|          | n. nodos         |          | % acierto        |          | n. nodos  |          | % acierto            |          | n.nodos   |          | % de acierto |          |
|          | $\bar{x}$        | $\sigma$ | $\bar{x}$        | $\sigma$ | $\bar{x}$ | $\sigma$ | $\bar{x}$            | $\sigma$ | $\bar{x}$ | $\sigma$ | $\bar{x}$    | $\sigma$ |
| sin poda | 13               | 3.47     | 96.52            | 1.56     | 8.42      | 2.8      | 94.88                | 2.8      | 3.21      | 0.8      | 98.42        | 1.73     |
| con poda | 11.51            | 3.12     | 96.59            | 1.46     | 7.25      | 2.16     | 94.64                | 4.33     | 3.02      | 0.14     | 98.63        | 1.55     |

Tabla 1: Resultados de la clasificación. Medias y desviaciones típicas.

A la vista de la tabla 1, cabe hacer algunos comentarios. Los resultados son prometedores, en el sentido de que hemos conseguido dos de los objetivos que buscábamos: ha disminuido la dispersión y también el número de categorías. Se han acercado a un óptimo que suponemos que estará en un menor número de nodos y un mayor acierto. Probablemente los casos de aumento en el número de nodos son debidos a una extrana configuración inicial. También se han dado algunos pocos casos de mucho descenso en el acierto, que son los que aumentan el valor de la dispersión en el acierto con reordenación. El siguiente paso es aplicar la reordenación sucesivas veces, alejándonos así del orden inicial aleatorio, y promocionando cada vez más el orden propuesto por el algoritmo.



A. Pedro Tejedor B. Jorge  
Martínez C. Eugenio F.  
Sánchez

Title Page

Contents



Go Back

Close

Quit

Page 19 of 24

## 5.1. Varias reordenaciones

Se ha lanzado el algoritmo recursivamente 50 veces, para observar su evolución, tanto en cuanto al número de categorías creadas como en cuanto al acierto que éstas tienen en el conjunto de test. Los resultados se pueden ver en la tabla 1, así como en la figura 3 en la que se puede apreciar la evolución de los 100 procesos.

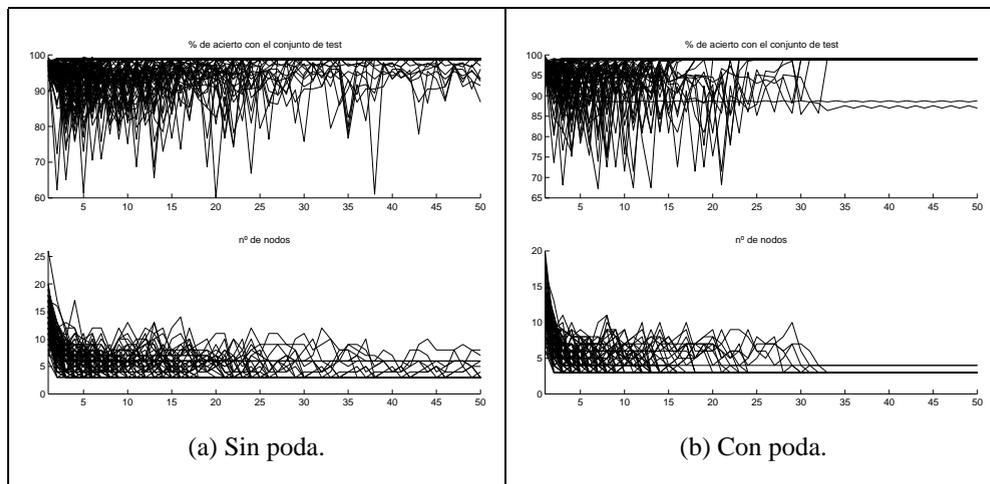


Figura 3: Evolución del algoritmo

Varios comentarios se pueden hacer a la vista de estos resultados. La reordenación ha sido capaz de bajar el número de clases de salida aun en el caso en que no había poda. El tanto por ciento de acierto en el test ha subido, con lo



A. Pedro Tejedor B. Jorge  
Martínez C. Eugenio F.  
Sánchez

Title Page

Contents



Go Back

Close

Quit

Page 20 of 24

que podemos suponer que son clases más generales que las que aparecían en los otros casos. La dispersión en el número de nodos ha bajado mucho. En el caso en que se ha usado la poda, todos los procesos han acabado en 3 categorías, salvo dos de ellos, que han acabado en cuatro. Son los mismos que han obtenido un resultado bajo en el test, que se puede apreciar en la figura 3(b). Por tanto, en algún caso patológico la reordenación puede darnos un resultado subóptimo. Quitando estos dos casos, podemos afirmar que la poda aumenta la estabilidad del algoritmo y acelera su convergencia. En el caso sin poda quedaban todavía algunos de los procesos que no habían alcanzado su mínimo estable después de 50 iteraciones.



A. Pedro Tejedor B. Jorge  
Martínez C. Eugenio F.  
Sánchez

Title Page

Contents



Go Back

Close

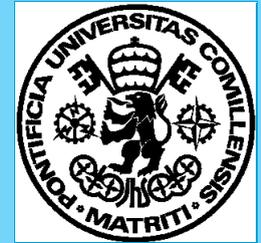
Quit

Page 21 of 24

## 6. CONCLUSIÓN

En este trabajo se ha revisado un popular algoritmo de clasificación automática que permite la extracción de reglas lingüísticas, Fuzzy-ARTMAP. Se han señalado algunas de sus dificultades, sobre todo la dependencia del orden de presentación de los ejemplos de entrenamiento, y se ha propuesto un método iterativo que permite aumentar la robustez y la estabilidad de las clases así creadas. Obtenemos en definitiva un sistema más robusto y fiable. Además, dado que el número de las clases de salida es menor, permite una extracción de reglas más limpia y fácil de comprender y evaluar por las personas. Todo lo anterior se ha comprobado en un ejemplo sencillo y por tanto de fácil evaluación, obteniendo unos resultados prometedores.

Queda abierto el camino de comprobación con clasificaciones más complejas, con las que habrá que afinar el criterio de reordenación de los ejemplos, para que se mantengan en la medida de lo posible las propiedades aquí conseguidas.



A. Pedro Tejedor B. Jorge  
Martínez C. Eugenio F.  
Sánchez

Title Page

Contents



Go Back

Close

Quit

Page 22 of 24

## Referencias

- [1] Carpenter, Gail A. and Grossberg, Stephen and Reynolds, John H. ART-MAP Supervised real-time learning and classification of nonstationary data by a self-organizing neural network. *Neural Networks* 1991 4 565–588 6
- [2] Carpenter, Gail A. and Grossberg, Stephen and Markuzon, Natalya and Reynolds, John H. and Rosen, David Fuzzy ARTMAP: A Neural Network Architecture for Incremental Supervised Learning of Analog Multidimensional Maps. *IEEE Transactions on neural networks* 1992 3(5) 698–713. 8, 11, 14
- [3] Carpenter, Gail A. and Tan, Ah-Hwee Rule extraction: from neural architecture to symbolic representation. *Connection Science* 1995 7(1) 3–27. 6, 12
- [4] Carpenter, Gail A. Distributed Learning, Recognition and Prediction by ART and ARTMAP Neural Networks. *Neural Networks* 1997 10(8) 1473–1494 6
- [5] Georgiopoulos, Michael and Fernlund, Hans and Bebis, George and Heillemann, Gregory L. Order of search in Fuzzy ART and Fuzzy ARTMAP: Effect of the choice parameter. *Neural Networks* 1996 9(9) 1541–1559. 6, 9, 10



A. Pedro Tejedor B. Jorge  
Martínez C. Eugenio F.  
Sánchez

Title Page

Contents



Go Back

Close

Quit

Page 23 of 24

- [6] Lim, Chee Peng and Harris, Robert F. Modified Fuzzy ARTMAP Approaches Bayes Optimal Classification Rates: An Empirical Demonstration. *Neural Networks* 1997 10(4) 755–774 6
- [7] Jorge Martínez Escribano Herramienta Informática para reconocimiento de números manuscritos mediante redes neuronales Fuzzy-ARTMAP. Proyecto Fin de Carrera. UPCO 1999 6
- [8] Williamson, James R. Carpenter, Gail A. Gaussian ARTMAP: A Neural Network for Fast Incremental Learning of Noisy Multidimensional Maps. *Neural Networks* 1996 19(5) 881–897 6



A. Pedro Tejedor B. Jorge  
Martínez C. Eugenio F.  
Sánchez

Title Page

Contents



Go Back

Close

Quit

Page 24 of 24