Computación y Sistemas Vol. 5 No. 2 pp. 89 - 95
© 2001, CIC - IPN. ISSN 1405-5546 Impreso en México

Clasificación de Patrones Temporales en Sistemas Dinámicos Mediante Boosting y Alineamiento Dinámico Temporal *

Temporal Patterns Classification in Dynamical Systems Through Boosting Dynamic Time Warping

Juan J. Rodríguez[1], Carlos J. Alonso[2] and Quilan Isaac Moro Sancho[2]
[1]Lenguajes y Sistemas Informáticos, Universidad de Burgos, España
[2]Grupo de Sistemas Inteligentes, Departamento de Informática, Universidad de Valladolid, España
e-mail : jjrodriguez@ubu.es, {calonso, isaac}@infor.uva.es

Artículo recibido en mayo 30, 2001; aceptado en septiembre 25, 2001

Resumen

Se propone un método novedoso de aprendizaje de patrones temporales, de interés en la diagnosis de procesos continuos en sistemas dinámicos. El método se basa en el uso de la familia de algoritmos de aprendizaje denominados boosting, que se caracterizan por mejorar el resultado de otro método de aprendizaje, llamado base, mediante la aplicación repetida del mismo. El método base de aprendizaje utilizado genera clasificadores muy simples, aunque específicos para el problema considerado. Dichos clasificadores se limitan a calcular la distancia del ejemplo en cuestión con otro de referencia, y comparar dicho valor con un umbral. La distancia utilizada es la proporcionada por el método de alineamiento dinámico temporal. La validación experimental del método se realiza mediante un conjunto de datos propuesto como banco de pruebas de sistemas de aprendizaje sobre patrones temporales en sistemas dinámicos. Los resultados experimentales, comparados con los conocidos para este conjunto de datos, son satisfactorios.

Palabras clave: Clasificación de Series Temporales, Boosting, Redes de Funciones de Base Radial, Aprendizaje Automático.

Abstract

A novel learning method is proposed for temporal patterns. This method is interesting for the diagnosis of continuous processes in dynamical systems. It is bases on the family of learning algorithms named boosting. These algorithms improve the results of other learning method, named based method, by means of the combination of the results obtained in different runnings. The uses base method generates very simple classifiers, but specialized in the considered problem. These classifiers only calculate the distance between the current example and another reference example, and this distance value is compared to a threshold. The distance used is obtained through the dynamic time warping method. The method has been validated experimentally using a data set that has been proposed as benchmark of machine learning methods for learning temporal patterns in dynamical systems. The experimental results, compared with the know results for this data set, are very satisfactory.

Keywords: Time Series Classification, Boosting, Radial Basis Function Networks, Machine Learning.

* Este trabajo ha sido financiado por el proyecto de la CYCIT TAP 99-0344 y el proyecto de la Junta de Castilla y León VA101/01

1 Introducción

El funcionamiento de cualquier planta industrial esta basado en las lecturas de un conjunto de sensores. La habilidad para identificar el estado de operación, o los eventos que están ocurriendo, a partir de los valores históricos de los sensores, es esencial en tareas como el control de supervisión, la detección y diagnosis de fallos, y el control de la calidad del proceso (Roverso, 2000).

Esta identificación puede realizarse mediante Sistemas Basados en Conocimiento, bien sea mediante desarrollos basados en la experiencia, como el descrito en (Alonso González and Rodríguez Diez, 1999), o con sistemas basados en aprendizaje computacional como el que se describe en este trabajo.

El método que se propone para clasificar patrones temporales esta basado en literales de similitud (comparación del valor de la distancia entre dos ejemplos con un umbral) y boosting (un método para la generación de agrupaciones, combinados de clasificadores) (Schapire, 1999)

Tomando como base los resultados sorprendentemente buenos de aplicar boosting sobre clasificadores base muy simples (denominados *stumps*, ya que son árboles con una sola decisión) para varios conjuntos de datos (Freund and Schapire, 1996), el método propuesto utiliza clasificadores base muy simples, únicamente un literal. El formato de un predicado es el siguiente:

$$\text{distancia}_{\leq}(\text{ Ejemplo, Referencia, Variable, Umbral })$$

Y un literal es un predicado posiblemente negado. El predicado es cierto si la distancia entre Ejemplo considerado y un ejemplo de Referencia, restringida a la Variable, es menor o igual que el Umbral.

El argumento Variable merece comentarios adicionales. Si se quisiera clasificar patrones con una sola variable (el valor de la variable en el tiempo forma una serie) sería innecesario. No obstante, en la diagnosis de sistemas continuos es necesario considerar varios sensores, y el problema a considerar es el de clasificación *multivariable*. Al introducir el

argumento de Variable en el predicado no se utiliza una distancia global entre ejemplos, sino que entre dos ejemplos hay tantas distancias como variables a considerar. Es necesario añadir que cuando hablamos de variable lo hacemos en el sentido, anteriormente expuesto. Desafortunadamente, desde el punto de vista de los métodos de aprendizaje, el concepto de variable es distinto.

El resto del artículo se organiza como sigue. La sección 2 describe el método de aprendizaje utilizado. Para ello, en un primer lugar se da una breve descripción del boosting, junto con algunos detalles sobre la implementación elegida. A continuación se describen los clasificadores base, basados en el uso de la distancia proporcionada por el alineamiento dinámico temporal. La validación experimental es objeto de la sección 3. Incluye una descripción del conjunto de datos utilizados así como una discusión sobre los resultados obtenidos. Finalmente, se concluye en la seccion 4, indicando aquellas tareas que se tiene previsto abordar como consecuencia del presente trabajo.

2 Descripción del Método

2.1 Introducción al Boosting

La agrupación de varios clasificadores, obtenidos utilizando un mismo método, es una manera natural de incrementar la precisión, con respecto a la obtenida con la utilización aislada de dichos clasificadores. Uno de los métodos más populares para crear estas agrupaciones de clasificadores es el *boosting* (Schapire, 1999). Este término engloba toda una familia de métodos, de la que ADABOOST es la variante más conocida.

Estos métodos trabajan asignando un peso a cada ejemplo. Inicialmente, todos los ejemplos tienen el mismo peso. En cada iteración, se construye un clasificador, denominado *base* o *débil*, utilizando algún método de aprendizaje, y teniendo en cuenta la distribución de pesos. A continuación, el peso de cada ejemplo se reajusta, en función de si el clasificador base le asigna la clase correcta o no. El resultado final se obtiene mediante voto ponderado de los clasificadores base.

En las siguientes secciones se indican algunos detalles sobre la versión concreta de boosting utilizada en este trabajo.

2.1.1 Asignación de Pesos a los Clasificadores Base

En (Schapire and Singer, 1998) se proponen varios métodos para seleccionar el peso (α) asociado a cada clasificador base. El mejor valor para α se obtiene minimizando

$$Z = \sum_i D(i) e^{-\alpha u_i}$$

Donde $D(i)$ es el peso del ejemplo x_i, $u_i = y_i h(x_i)$, $y_i \in \{-1, +1\}$ es la clase del ejemplo y $h(x_i)$ es la confianza asignada por el clasificador base al ejemplo. Para la variante original de ADABOOST, esta expresión se aproxima (restringiendo $h(x_i)$ a $[-1, +1]$) por

$$Z \leq \sum_i D(i) \left(\frac{1+u_i}{2} e^{-\alpha} + \frac{1-u_i}{2} e^{\alpha} \right)$$

Y el mínimo α para esta expresión se seleccionaba analíticamente.

Sin embargo, sugerían la posibilidad de utilizar otras cotas superiores, y en este trabajo se utiliza la cota

$$\begin{aligned} Z \leq & \sum_{i:u_i \geq 0} D(i)(u_i e^{-\alpha} - u_i + 1) \\ & + \sum_{i:u_i < 0} D(i)(-u_i e^{+\alpha} + u_i + 1) \end{aligned}$$

que proporciona una aproximación más ajustada.

2.1.2 Problemas Multiclase

El planteamiento inicial del algoritmo ADABOOST considera problemas binarios, con sólo dos clases. Hay varios métodos para extender este algoritmo a situaciones en las que se trabaja con más de dos clases, como ADABOOST.MH y ADABOOST.MR (Schapire and Singer, 1998). Sin embargo, estos métodos requieren que los clasificadores base sean multiclase, lo que no les hace adecuados para nuestro caso, en el que tenemos clasificadores base binarios.

Por otro lado, ADABOOST.OC (Schapire, 1997) es un método para extender ADABOOST a problemas multiclase que utiliza clasificadores base binarios. La idea principal de esta variante es, en cada iteración de boosting, seleccionar un subconjunto del conjunto de clases, y entrenar un clasificador base binario en el que todos los ejemplos de cualquiera de las clases en el subconjunto se consideran positivos y el resto de los ejemplos se consideran negativos.

La implementación utilizada en nuestro caso está basada en una variante adicional de ADABOOST.OC, denominada ADABOOST.ECC (Guruswami and Sahai, 1999).

2.2 Clasificadores Base

2.2.1 Selección de Literales

El funcionamiento del sistema de aprendizaje base es el siguiente. En primer lugar, se seleccionan, aleatoriamente, varios ejemplos, como referencias. El número de ejemplos de referencia considerado (r), es un parámetro. En la implementación actual se selecciona el mismo número de ejemplos positivos y negativos, aunque sería posible utilizar distintos valores o incluso no utilizar ejemplos negativos.

Para cada ejemplo de referencia, se calcula la distancia al resto de ejemplos (e) de entrenamiento . El tiempo necesario para este proceso es de $e\,t(n)$, donde $t(n)$ es el tiempo necesario para calcular las distancias entre dos ejemplos con n atributos. En el caso de la distancia euclidea es $t(n) \in O(n)$.

A continuación, se selecciona el mejor umbral para las distancias calculadas, de manera similar a como se selecciona cuando se expande un nodo en la construcción de árboles de decisión (Quinlan, 1993). Para esto se ordenan las distancias (en un tiempo de $e \lg e$). Se recorren los ejemplos, en el orden determinado por la distancia, en sentido ascendente, llevando en cuenta el número (y peso) de los ejemplos, positivos y negativos, cuya distancia es menor que la del ejemplo actual. Para cada valor de distancia presente, se calcula el error cometido al seleccionar este umbral. Este cálculo puede realizarse en $O(1)$, porque sólo es necesario calcular una función de los pesos de los ejemplos positivos y negativos a la izquierda y a la derecha del umbral. Para e ejemplos, una vez que los valores están ordenados, la selección del mejor umbral puede hacerse en $O(e)$. Este tiempo es inferior al necesario para ordenar las distancias, $O(e \lg e)$. Por tanto, el tiempo necesario para la selección del clasificador base es $re(t(n) + \lg e)$.

2.2.2 Grados de Confianza

Dado un literal, sobre un ejemplo podemos tener dos resultados, cierto o falso, que para el algoritmo de boosting se consideran +1 y -1. No obstante, el método de boosting puede trabajar con clasificadores base que devuelven valores reales, denominados grados de confianza. En este caso, la cantidad de información intercambiada entre el algoritmo de aprendizaje base y el algoritmo de boosting es mayor, puesto que la salida del clasificador base no esta limitada a positivo o negativo, sino que es un número real que indica el grado de confianza.

El método seguido para obtener grados de confianza continuos a partir de literales consiste en trasformar estos en funciones de base radial, RBF. Dado un literal

distancia$_{\leq}$(Ejemplo, Referencia, Variable, Umbral)

la función de base radial seleccionada es

$$h(x) = 2 \exp\left(- \left(\frac{d_V(x,c)}{t} \right)^2 \ln 2 \right) - 1$$

donde x es el Ejemplo, c el ejemplo de Referencia, u el Umbral y d_v la distancia restringida a la Variable v.

Esta función tiene las siguientes propiedades:

- $h(c) = 1$
- $h(x) = 0$ if $d_v(x,c) = t$
- $-1 \leq h(x) \leq 1$, supuesto que $d_v(x,c) \geq 0$
- Esta función desciende monótonamente con respecto a $d_v(x,c)$

Si el literal aparece negado, se multiplica esta función por -1.

Estas funciones son funciones de base radial (Orr, 1996), y su combinación lineal es un red de funciones de base radial, *RBFN*.

2.2.3 Alineamiento Dinámico Temporal

El Alineamiento Dinámico Temporal (*Dynamic Time Warping, DTW*) ajusta (expande o contrae fragmentos) una serie temporal a otra de referencia de modo que una función de distancia sea minimizada, utilizando un algoritmo de programación dinámica (Berndt and Clifford, 1996). El valor mínimo obtenido se puede considerar como una distancia entre las dos series.

Respecto a la distancia euclídea aporta la ventaja de que las dos series no tienen que tener la misma longitud, pero la principal aportación es que es más robusta a alteraciones (expansiones, compresiones) en el eje temporal.

3 Validación Experimental

3.1 Conjunto de Datos

El conjunto de datos utilizado se introduce en (Roverso, 2000). Es un conjunto de datos generado artificialmente que se propone como banco de pruebas para sistemas de clasificación de patrones temporales en la industria de procesos.

Hay cuatro variables (series), cada una de las cuales puede presentar dos comportamientos. La figura 1 muestra ejemplos de estos comportamientos, que se describen a continuación.

- Para la primera variable, en los dos casos, tiene forma de escalón ascendente. Lo que diferencia un comportamiento del otro es la amplitud de este incremento.
- En la segunda variable, en ambos casos aparece un decremento pronunciado de los valores, seguido de una recuperación lenta hasta los valores iniciales. En uno de los casos, el decremento es precedido por un pico de valores altos.
- En la tercera variable aparece el escalón ascendente. En uno de los comportamientos en la parte alta del escalón aparece una pequeña oscilación.
- Para la cuarta variable, uno de los comportamientos es de escalón ascendente, el otro empieza de la misma manera, pero los valores de la variable a partir de un punto dado empiezan a decrecer hasta alcanzar los valores originales.

Combinando los posibles comportamientos de cada variable, se obtienen 16 casos. Cada caso se corresponderá con una clase distinta. Este conjunto de datos se diseño así con el objetivo de que fuera difícil: para cada clase hay otras cuatro con las que sólo se diferencia en el comportamiento de una de las variables.

La longitud base considerada fue de 300 puntos. Las modificaciones realizadas sobre los prototipos de cada clase, para generar los ejemplos fueron:

- Variación de amplitud: $\pm 30\,\%$

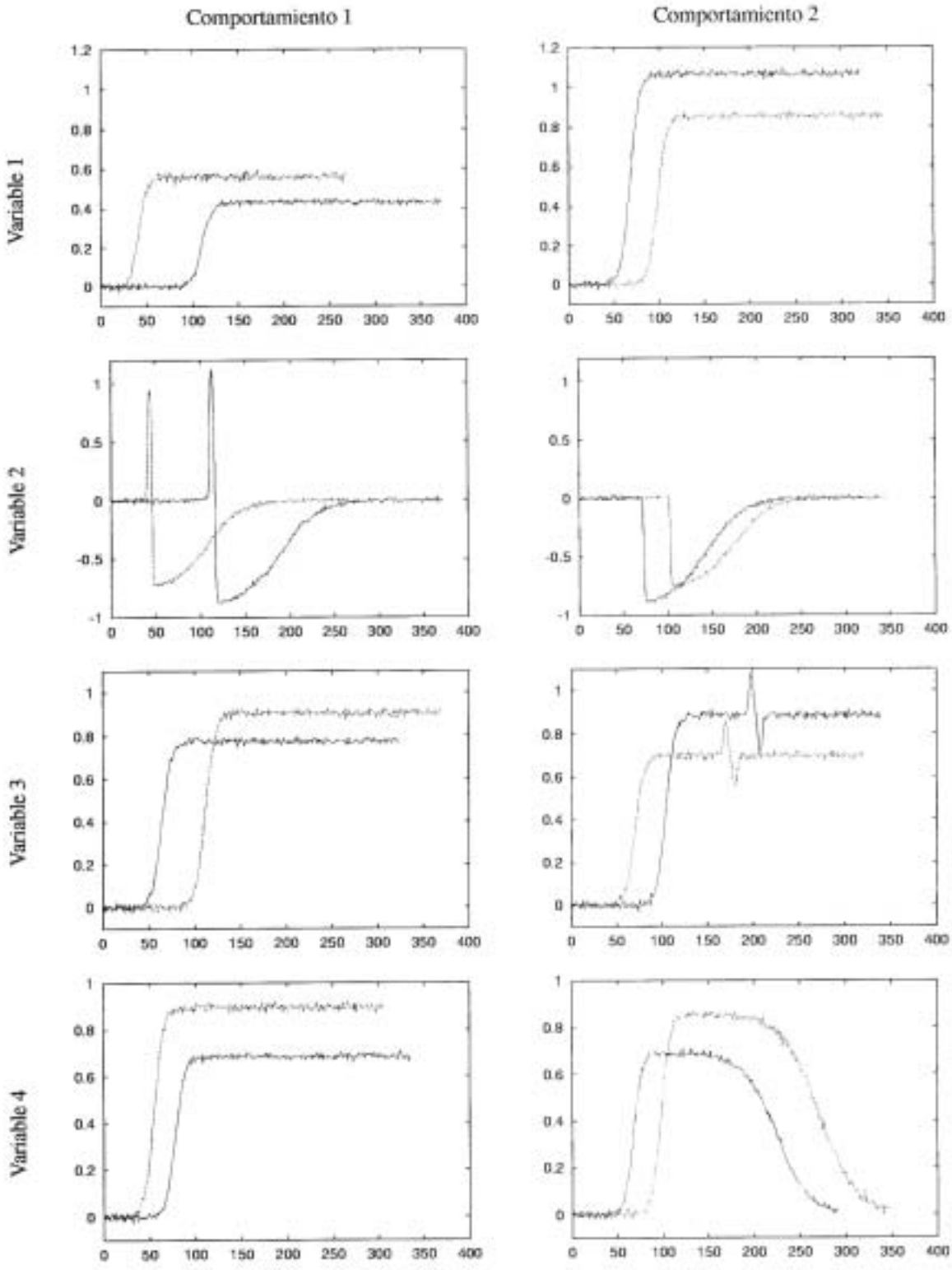

Figura 1: Comportamientos de las distintas variables. En cada una de las gráficas se muestran dos series correspondientes al mismo comportamiento de una de las variables.

- Retardo: hasta el 20 %.
- Variación en el eje temporal: ±20 %
- Ruido: ±1 % gaussiano.

Se generaron 1600 ejemplos, 100 de cada clase. De ellos, una mitad (50 de cada clase) es de entrenamiento y la otra para prueba. El rango de longitudes en los ejemplos va desde 268 hasta 394.

3.2 Resultados

El experimento, para cada una de las dos variantes, se repitió 5 veces. Los resultados obtenidos se muestran en la tabla 1 y en la figura 2. Sobre estos resultados se pueden destacar dos aspectos.

- En estos resultados no hay problemas de sobreajuste, las curvas son claramente descendentes. Sin embargo, aunque los métodos de boosting son bastante robustos al sobreajuste, no son inmunes; por tanto no se pude extrapolar esta circunstancia a cualquier conjunto de datos.
- A partir de un punto dado, los resultados de la variante continua son mejores que los de la variante discreta.

El error obtenido en (Roverso, 2000), utilizando redes neuronales recurrentes sobre wavelets, es de 1.4 %. Dicho valor es mayor que nuestros resultados para la versión continua con 600 iteraciones. Por otro lado, en ese trabajo, además del error del 1.4 %, un 4.5 % de los ejemplos se dejan sin clasificar, no se les asigna ninguna clase. La suma de estos dos valores (5.9) también es mayor que nuestros resultados en la versión discreta (2.5)

El resultado obtenido con la versión continua es mejor que el resultado obtenido con la versión discreta. Ahora bien, ¿es esta diferencia significativa? Para intentar resolver a esta pregunta, se consideran los resultados obtenidos por las dos variantes en cada una de las cinco ejecuciones, utilizando el test de McNemar (Dietterich, 1998). La tabla 2 compara los resultados obtenidos por las dos variantes. Todos los resultados obtenidos con la versión discreta son peores que los obtenidos con la versión continua. En 4 de las 5 ejecuciones esa diferencia es claramente significativa.

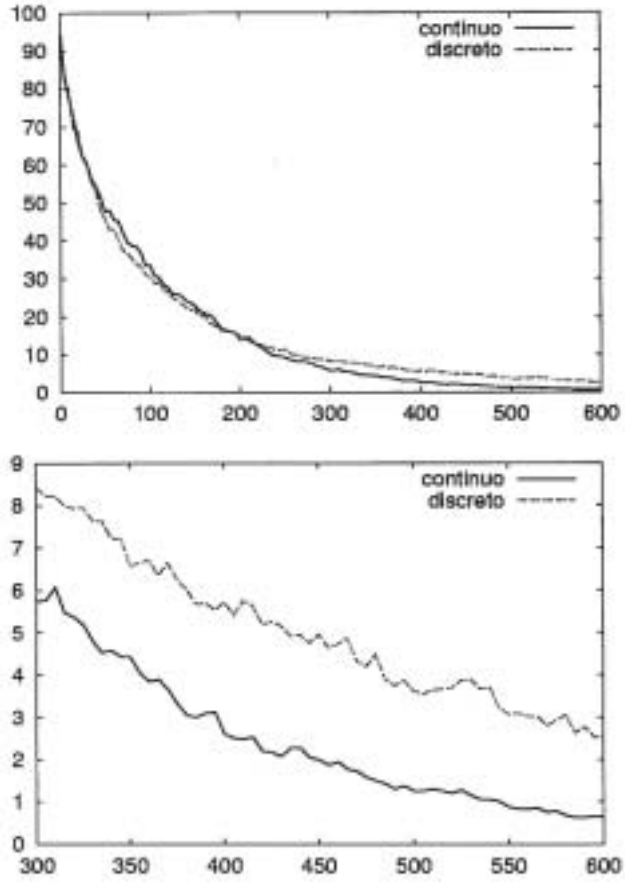

Figura 2: Gráficas de los resultados. La primera gráfica muestra los resultados para las 600 iteraciones. La segunda sólo muestra las últimas 300 iteraciones. En cada gráfica se muestran, para las dos variantes, continua y discreta, los valores medios de error obtenidos para ese número de iteraciones.

4 Conclusiones y Trabajo Futuro

Se ha presentado un método novedoso para la clasificación de patrones temporales. Este método se basa, por un lado, en el uso del método de boosting, que es un marco general en el que mejorar los resultados de un método de aprendizaje mediante la repetición del mismo. Y por otro, en el uso de clasificadores base muy simples, pero específicos para el tipo de problemas a considerar. Estos clasificadores base calculan la distancia del ejemplo en cuestión a otro de referencia, y comparan dicho valor con un umbral. Gran parte del éxito del método se debe al uso de una distancia, proporcionada por el alineamiento dinámico temporal, adecuada para este tipo de datos.

Se introduce una mejora en el método mediante el uso de factores de confianza. Para la asignación de dichos factores, a partir de cada literal de distancia se obtiene una función de base radial. La combinación lineal de estas funciones proporcionada por el boosting es una red de funciones de base radial. Aunque con la particularidad, en este caso, de utilizar la distancia DTW.

Los resultados experimentales obtenidos, son claramente favorables con respecto a los previamente conocidos para este conjunto de datos. No obstante, admiten dos críticas:

- El conjunto de ejemplos utilizado ha sido generado de manera artificial. Sin embargo hay que tener en cuenta que este conjunto de datos se propone precisa-

Iteraciones	50	100	150	200	250	300	350	400	450	500	550	600
Discreto												
Error medio	45.73	30.40	21.58	14.88	11.43	8.43	6.58	5.73	4.95	3.63	3.08	2.50
Desviación	4.62	2.72	5.11	4.68	4.54	2.73	2.20	1.49	1.70	1.33	1.11	1.03
Continuo												
Error medio	47.63	33.70	22.83	14.05	9.20	5.75	4.43	2.60	2.00	1.25	0.88	0.65
Desviación	3.15	2.00	4.84	4.71	4.05	1.94	1.83	0.93	1.02	0.85	0.34	0.26

Cuadro 1: Resultados experimentales.

Experimento:		1	2	3	4	5
Ejemplos mal clasificados	Discreto	29	22	25	16	8
	Continuo	6	3	7	3	7
	Ambos	0	0	1	2	0
Significancia:		0.00012	0.00016	0.00143	0.00098	1.00000

Cuadro 2: Comparación entre los dos métodos. Para cada uno de los 5 experimentos se muestra el número de ejemplos (de prueba) mal clasificados por la variante discreta, la continua y cuántos son clasificados erróneamente por las dos. Con esos tres valores se obtiene el valor de significancia. La diferencia es más significativa cuanto más pequeña.

mente para comparar distintos sistemas de clasificación de este tipo de datos (Roverso, 2000). Además, este método sí que ha sido probado sobre datos reales (Rodríguez Diez and Alonso González, 2001), aunque ajenos a problemas de diagnosis.

- El número de iteraciones de boosting necesario para obtener buenos resultados es grande. El problema principal no es el tiempo de aprendizaje, sino el tiempo necesario para clasificar. Este tiempo viene dado principalmente por el número de distancias a calcular. Por tanto, uno de los temas de investigación pendientes es si es posible obtener resultados similares utilizando un menor número de ejemplos de referencia.

El presente trabajo se centra en como obtener clasificadores. El cómo incluir estos clasificadores en un sistema de supervisión es materia de futuros trabajos.

Referencias

Alonso González, C. J. and **Rodríguez Diez, J. J.** "A graphical rule language for continuous dynamic systems". In Mohammadian, M., editor, *Computational Intelligence for Modelling, Control and Automation: CIMCA-99*, 1999, pp. 482–487. IOS Press.

Berndt, D. and **Clifford, J.** "Finding patterns in time series: a dynamic programming approach". In Fayyad, U., Piatetsky-Shapiro, G., Smyth, P., and Uthurusamy, R., editors, *Advances in Knowledge Discovery and Data Mining*, pp. 229–248. AAAI Press / MIT Press.

Dietterich, T. G. "Approximate statistical tests for comparing supervised classification learning algorithms". *Neural Computation*, Vol. 10, No. 7, 1998, pp. 1895–1924.

Freund, Y. and **Schapire, R.** "Experiments with a new boosting algorithm". In *13th International Conference om Machine Learning (ICML-96)*, 1996, pp. 148–156.

Guruswami, V. and **Sahai, A.** "Multiclass learning, boosting, and error-correcting codes". In *12th Annual Conference on Computational Learning Theory (COLT 1999)*, 1999. ACM.

Orr, M. J. "Introduction to radial basis function networks". Technical report, 1996. `http://www.anc.ed.ac.uk/~mjo/papers/intro.ps.gz`.

Quinlan, J. R. *C4.5: programs for machine learning*. Machine Learning. Morgan Kaufmann, San Mateo, California, 1993.

Rodríguez Diez, J. J. and **Alonso González, C. J.** "Learning classification RBF networks by boosting". In Kittler, J. and Roli, F., editors, *2nd International Workshop on Multiple Classifier Systems (MCS 2001)*, Lecture Notes in Computer Science, 2001, pp. 43–52. Springer.

Roverso, D. "Multivariate temporal classification by windowed wavelet decomposition and recurrent neural networks". In *3rd ANS International Topical Meeting on Nuclear Plant Instrumentation, Control and Human-Machine Interface*, 2000.

Schapire, R. E. "Using output codes to boost multiclass learning problems". In *14th International Conference on Machine Learning (ICML-97)*, 1997, pp. 313–321.

Schapire, R. E. "A brief introduction to boosting". In Dean, T., editor, *16th International Joint Conference on Artificial Intelligence (IJCAI-99)*, 1999, pp. 1401–1406. Morgan Kaufmann.

Schapire, R. E. and **Singer, Y.** "Improved boosting algorithms using confidence-rated predictions". In *11th Annual Conference on Computational Learning Theory (COLT 1998)*, 1998, pp. 80–91. ACM.

Juan J. Rodríguez Diez *es Diplomado en Informatica, 1992, y Licenciado en Informática, 1994, por la Universidad de Valladolid. Es profesor del Área de Lenguajes y Sistemas Informáticos, en la Universidad de Burgos.Sus temas de investigación se encuadran dentro del área del Aprendizaje Automático, más en concreto la Clasificación de Series Temporales y los Sistemas de Combinación de Clasificadores.*

Carlos J. Alonso González *es Licenciado en Ciencias, 1985, y doctor en Ciencias Físicas, 1990, en ambos casos por la Universidad de Valladolid. Ha participado en numerosos proyectos relacionados con la supervisión y diagnosis de procesos industriales continuos. En la actualidad es Secretario del Centro de Tecnología Azucarera y Profesor Titular de Universidad en el Departamento de Informática. Sus principales temas de investigación son los Sistemas Basados en Conocimiento para la Supervisión y Diagnosis de Procesos Continuos, Diagnosis Basada en Modelos de Sistemas Dinámicos, Ingeniería del Conocimiento y Aprendizaje Automático.*

Quilian I. Moro Sancho *es Licenciado en Ciencias Físicas por la Universidad de Valladolid en 1989. Doctorado por la Universidad de Valladolid en 2000, con la Tesis titulada «Una aportación a la Predicción Meteorológica Basada en Técnicas Conexionistas». Sus principales temas de investigación son Aplicación de las Redes Neuronales Artificiales, Reconocimiento de Patrones, Combinación de Redes Neuronales Artificiales.*