A New Kernel to use with Discretized Temporal Series

Un Nuevo Kernel para usar con Series Temporales Discretizadas

Luis González Abril[a], Francisco Velasco Morente[a], Juan Antonio Ortega Ramírez[b] and Francisco Javier Cuberos García Vaquero[c]

[a] Department of Applied Economics I, University of Seville (Spain)
[b] Department of Computer Science, University of Seville (Spain)
[c] Department of Planificación-Radio Televisión de Andalucía, Seville (Spain)
luisgon@us.es, velasco@us.es, ortega@lsi.us.es, fjcuberos@rtea.es

Article received on December 14, 2005; accepted on September 25, 2007

Abstract

In this paper a new Kernel, from statistical learning theory is proposed to work with symbols chains (words) obtained from a discretization procedure of a continuous features. Although the exact definition of the discretization is not strictly necessary, there must always exist either, a measure of distance or a similarity between symbols in a certain alphabet (a set of symbols). This kernel is applied on a set of television shares obtained from the seven main television stations in Andalusia (Spain). A comparative study for classification purposes is done, and the associated parameter selection is studied. Finally, it must be mentioned that this kernel has certain implications in the type of considered similarity that will be studied in further researches. The small influence of the λ parameter in identification tasks must also be discussed.

Keywords: Kernels, Discretization, Intervals Distance.

Resumen

En este artículo, un nuevo kernel (núcleo), procedente de la Teoría del aprendizaje Estadístico, es propuesto para trabajar con cadenas de símbolos obtenidos a través de un proceso de discretización de una variable continua. Aunque para la exacta definición de discretización no es estrictamente necesaria, siempre debe existir una medida de distancia o una medida de similitud entre símbolos en un determinado alfabeto (conjunto de símbolos). Este kernel es aplicado sobre un conjunto de repartos de audiencias en la televisión obtenido de las siete principales cadenas de televisión en Andalucía (España). Una comparativa con objeto de llevar a cabo una clasificación es realizada y la selección de parámetros es estudiada. Finalmente, mencionar que este kernel tiene ciertas implicaciones en el tipo de similaridad considerada las cuales serán estudiadas en futuras investigaciones. La poca influencia del parámetro λ en las tareas de identificación también debe ser analizada.

Keywords: Kernels, Discretización, Distancia Intervalar.

1 Motivation

Automated processing and knowledge extraction from data is an important task performed by machine learning algorithms. Hence, the generation of classification rules from class-labelled examples is possible. Instances can be described by a set of numerical, nominal, or continuous features. Several of these algorithms are expressly designed to handle numerical or nominal data, other algorithms perform better with discrete–values features, despite the fact that they can also handle continuous features (Kurgan and Cios, 2004). Meanwhile a certain number of algorithms developed in the machine learning community focus on learning from nominal feature spaces. Real–world classification includes patterns with continuous features where such algorithms can not be applied, unless the continuous features are firstly discretized. Discretization is the process of transforming a continuous attribute into a finite number of intervals associated with a discrete, numerical value–a number, symbol or letter. This is the usual approach for learning tasks that use mixed–mode–continuous and discrete–data. The discretization process is developed in two stages: given the range of values for the continuous attribute, first the number of discrete intervals is found, then, the width or boundaries for the intervals.

In (Macskassy et al, 2003) it was shown than even on purely numerical-valued data, results for text classification on the derived text-like representation outperforms the more naive numbers-as-tokens representation and, more importantly, is competitive with mature numerical classification methods such as C4.5, Ripper and SVM. The most straightforward way is to treat each number that a feature may take on as a distinct "word", and proceed with the use of a text classification method using the combination of true words and tokens-for-numbers words. However, his makes the numbers 1 and 2 as dissimilar as the numbers 1 and 100–all three values are unrelated tokens in the classification methods. An approach to applying text-classification methods problems with numerical-valued features would be desirable so that the distance between such numerical values can to be discerned by the classification method. Most of the methods translating a continuous feature into symbols –letters– in order to deal with texts –letters chains– lose part of their efficiency since they are not designed for this end. The kernel proposed in this paper is specifically designed to work with letters chains coming from a discretization process of a continuous feature and it highlights the properties of these features. To cope with the effectiveness of this kernel, it will be used on words from a dictionary where a distance exists between letters of the alphabet. The kernel was firstly proposed to compare time series that had been converted into symbol chains –words– (Cuberos et al, 2003, Cuberos et al, 2004). Thus, the similarity measure between words quantified a distance between original time series. Figure 1 shows an example of a partial typified curve with their derivative values and the assigned label to each transition between adjacent values.

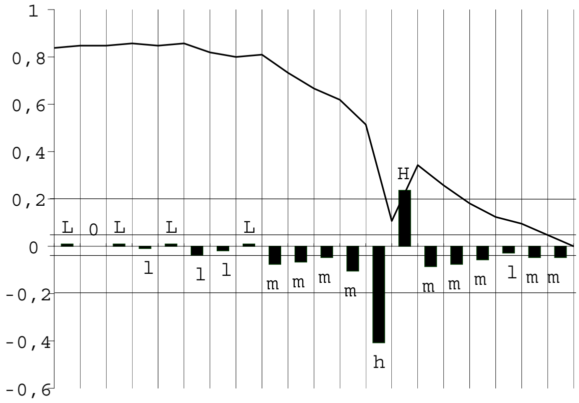

Fig. 1. Sample of translation from a time serie to letters chain

The rest of this paper is structured as follows: first, both a kernel and a distance between finite intervals are defined. Distance is used to define a real function measuring the similarity between two words and if words have the same length, this function is a Kernel because it fulfils the Mercer condition. Next, one example about classification rules is developed. Finally, the conclusions and ideas for future works are enumerated.

2 Interval distance from a Kernel

In essence the goal in the construction of kernel functions is to guarantee the existence of an application ϕ, defined from the working set, Ξ to a vectorial space endowed with a dot product, Φ. From this function ϕ the kernel function is defined, denoted $k(\cdot,\cdot)$, over pairs of elements of the working set as the dot product of their transformations into the feature space, $k(\cdot,\cdot)=\langle\phi(\cdot),\phi(\cdot)\rangle_{\Phi}$, where $\langle\cdot,\cdot\rangle_{\Phi}$ is denoted a dot product. The kernel function $k(\cdot,\cdot)$ let us establish similarities between the original elements from their transformed ones, so a distance between the points in the input space can be defined. It must be considered, when elaborating a similarity and distance measure, that the ϕ application must be able to highlight the essential characteristics of the initial set of elements.

Following the ideas presented in (González et al, 2004), let $I = \{(c-r, c+r) \subset \mathbb{R} : c \in \mathbb{R}, r \in \mathbb{R}\}$ be the family of all the open intervals contained in the real line$\mathbb{R}$[1] of finite dimension. A function $\phi_1 : I \to \mathbb{R}^2$ is defined as: $\phi_1(I) = P\begin{pmatrix} c \\ r \end{pmatrix}$ and the kernel $k(\cdot,\cdot)$ and a distance d_1^2 between intervals are:

$$k(I_1, I_2) = (c_1 \;\; r_1)\; S \begin{pmatrix} c_2 \\ r_2 \end{pmatrix} \;, \quad \mathrm{d}_1^2(I_1, I_2) = (\Delta c \;\; \Delta r)\; S \begin{pmatrix} \Delta c \\ \Delta r \end{pmatrix} \qquad (1)$$

where $I_1 = (c_1 - r_1, c_1 + r_1)$, $I_2 = (c_2 - r_2, c_2 + r_2)$, $\Delta\text{c}{=}c_2 - c_1$, $\Delta\text{r}{=}r_2 - r_1$, and P must be a non singular matrix ($S = P^t P$). Thus, the discretization of a continuous feature in symbols, usually letters, representing different intervals, allows us to use as a distance between symbols, the distance defined between intervals as it will be showed.

3 Kernel

From this point, we always consider that the symbols are letters (A, B,…) because the ordinal scale is reflected in the alphabetical order. Let $\mathrm{A} = \{A_1, A_2, \cdots, A_n\}$ be an alphabet of ℓ letters and let P be a set of the words that can be built from this alphabet. Let $P1 = P1_1 P1_2 \cdots P1_n$ and $P2 = P2_1 P2_2 \cdots P2_m$ be words from P where $P1_i, P2_j \in \mathrm{A}$ and $n \geq m$. A map K_λ is defined as:

$$K_\lambda(P1, P2) = \max\left\{ \sum_{i=1}^{m} \lambda^{d^2(P1_{i+k}, P2_i)}, k = 0, \cdots, n-m \right\}$$

where $0 < \lambda < 1$ and $d(\cdot,\cdot)$ is a distance between letters.

Property 1: For all $P1, P2 \in P$ and $0 < \lambda_1 < \lambda_2 < 1$, then: $K_{\lambda_1}(P1, P2) \leq K_{\lambda_2}(P1, P2)$.

Property 2: For all $P1, P2 \in P$ and $0 < \lambda < 1$, then: $K_\lambda(P1, P2) \leq m$. This upper bound is attained: If $P2 = P1_1 P1_2 \cdots P1_m$, then $K_\lambda(P1, P2) = m$.

Property 3: Let $r = \max_{ij} d(A_i, A_j)$, with $A_i, A_j \in \mathrm{A}$. For all $P1, P2 \in P$ and $0 < \lambda < 1$, then: $m^{\lambda^{r^2}} \leq K_\lambda(P1, P2)$. This lower bound is attained: Let $A = A_i$ and $B = A_j$ be such that $d(A, B) = r^2$. If $P1 = AA \cdots A$ and $P2 = BB \cdots B$ with size of $P1$, n, and size of $P2$, m, then $m^{\lambda^{r^2}} = K_\lambda(P1, P2)$. Thereby, for all $0 < \lambda < 1$: $\quad m^{\lambda^{r^2}} \leq K_\lambda(P1, P2) = m, \;\; \forall P1, P2 \in P$.

Property 4: Let A be an alphabet obtained from a discretization process of a continuous feature and $P = \{ P_1, P_2, \cdots, P_n, \; P_i \in \mathrm{A}\}$ the set of all the words having length n, then $K_\lambda(P1, P2) = \sum_{i=1}^{n} \lambda^{d^2(P1_i, P2_i)}$ is a kernel.

[1] In default, we are working with open intervals, but it is possible to translate the study to closed intervals naturally.

Proof: Let A be an alphabet obtained from a discretization process of a continuous feature into intervals. We consider the distance between intervals previously defined and a map φ from A to $\mathbb{R}^2$ is defined in the following way: Each interval (c–r , c+r) from the discretization process is denoted by a letter from alphabet A . Thus we consider the map ϕ_1 defined from I to $\mathbb{R}^2$ and consider the composition between this and the intervals distance. A new map is defined as $\phi : \mathrm{A} \to \mathbb{R}^2$ such that:

$$\phi(A) = P\begin{pmatrix} c \\ r \end{pmatrix}, \quad \forall \mathrm{A} \in \mathrm{A}$$

where P is a 2×2 matrix. It is defined a map $k_1 : \mathrm{A} \times \mathrm{A} \to \mathbb{R}$ such that: $K_1(A,B) = \langle \phi(A), \phi(B) \rangle$. It is a kernel function by construction, because it is a dot product. Therefore: $d^2(A,B) = \langle \phi(A) - \phi(B), \phi(A) - \phi(B) \rangle$ is a pseudo-distance between words.

Applying Corollary 3.13 (page 43) in (Cristianini and Shawe-Taylor, 2000), instead of the exponential function e^{-x} to the function $\left(\frac{1}{\lambda}\right)^{-x}$, with $0 < \lambda < 1$, it is true that the map $k_2 : \mathrm{A} \times \mathrm{A} \to \mathbb{R}$ defined in the following way: $K_2(A,B) = \lambda^{\|\phi(A)-\phi(B)\|^2} = \lambda^{d^2(A,B)}$ is a Kernel function. Now, if we consider the map $K_\lambda : P \times P \to \mathbb{R}$ such that:

$$K_\lambda(P1,P2) = \sum_{i=1}^{n} \lambda^{d^2(P1_i, P2_i)}$$

and using the proposition 3.12 in (Cristianini and Shawe-Taylor, 2000), it is true that $K_\lambda(\cdot,\cdot)$ is a kernel function. +

The λ parameter measures the importance that $K_\lambda(\cdot,\cdot)$ give to matching symbols versus the comparison of different symbols. For coincident symbols the value is always 1. This is depicted in Figure 2.

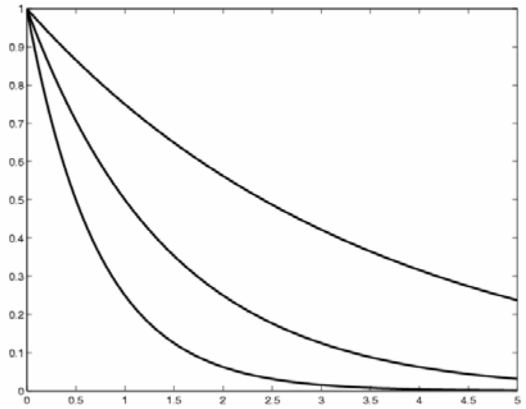

Fig. 2. Graph representation of function f (x) = λ^x for several values of λ

In this language each word has a meaning since it represents a whole interval of values. For this reason, we should ask ourselves which are the characteristics we want to take into account in each word of the language to be able to extract meaning from them. The function $K_\lambda(\cdot,\cdot)$ considers the following ones: i) The order of the letters into each word, ii) Comparison letter by letter, and iii) The size of the words.

3.1 Generalized similarity

Let $P1$ and $P2$ be two words of the same length n from the set Π. In the definition of similarity between words, $K_\lambda(P1,P2)=\sum_{i=1}^{n}\lambda^{d^2(P1_i,P2_i)}$, all the letters have the same interest. It is possible to generalize this similarity by weighting each letter in such a form that the sum of the weights is equal to n.

Let $w_1,w_2,\cdots,w_n \in$ □ be scalar numbers accomplishing $w_i \geq 0$ and $\sum_{i=1}^{n} w_i = n$. The generalized similarity can be defined in two different ways:

$$K_\lambda^1(P1,P2)=\sum_{i=1}^{n}\lambda^{w_i\, d^2(P1_i,P2_i)},\ K_\lambda^2(P1,P2)=\sum_{i=1}^{n} w_i\,\lambda^{d^2(P1_i,P2_i)}$$

It is not difficult to prove that both are kernels2, however the second one has a more intuitive meaning for the weights. Also, using the properties of the exponential function we have:

$$K_\lambda^1(P1,P2)=\sum_{i=1}^{n}\lambda^{w_i\, d^2(P1_i,P2_i)}=\sum_{i=1}^{n} w_i^{'}\lambda^{d^2(P1_i,P2_i)}$$

where $w_i^{'}=\lambda^{(w_i-1)d^2(P1_i,P2_i)}$. Although it is no necessarily true that $\sum_{i=1}^{n} w_i^{'} \neq n$. For this we propose as a generalization of similarity the function $K_\lambda^2(\cdot,\cdot)$. In Figure 3, several examples of weighting can be observed.

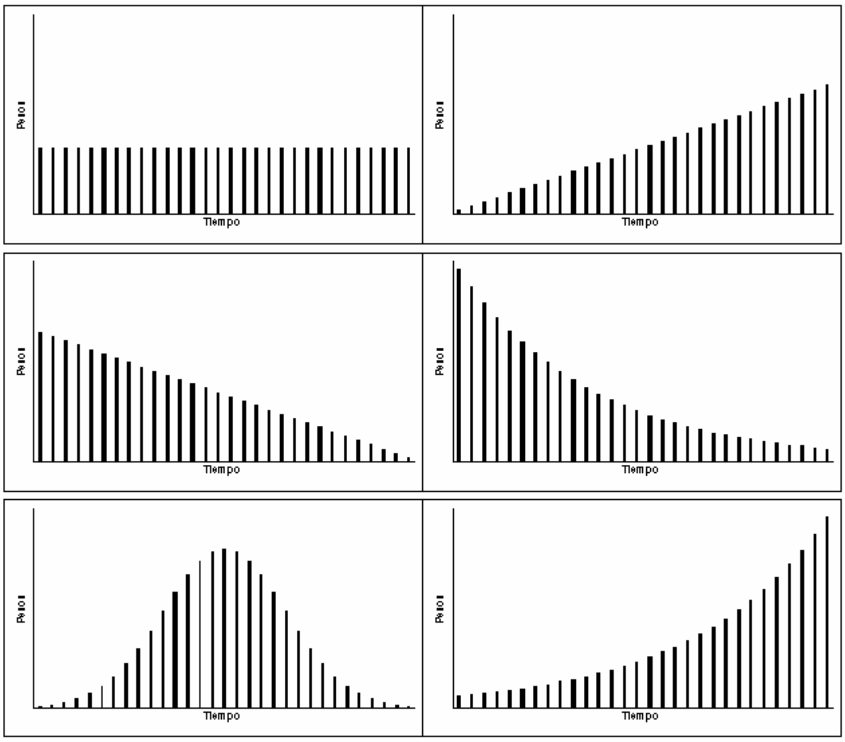

Fig. 3. Different weighting shapes to build generalized similarities

[2] The sum and the product of kernels is a kernel (Cristianini and Shawe-Taylor, 2000).

3.2 Evaluation and Identification

We are going to evaluate the quality of a discretization method, that is, its ability to identify the class to which a new series from the work set belong. The test tries to identify every series by using the nearest neighbour algorithm. The label of the most similar learning series to the new series is checked in front of the label in this series, checking thereby if the system chooses the right label. For this test every discretization method is applied to all the series. The application of the methods consists on transforming the series into symbol strings and calculating the similarity between every pair by means of the distance defined in (1) Once all the results are obtained for each method in every attempt, by changing the learning and test subsets the best method for the current data set is selected. After the discretization method has been selected, it can be applied to all the series in the learning set, allowing to obtain the final set of interval landmarks. Finally, the system notifies the user that the end of the learning process has been reached. Now the user can present a set of new unlabelled series, the work set, and obtains an answer from the system.

4 Implementation

An example of the classification rule is developed. Data to be considered is a set of television shares from the seven main television stations in Andalusia, Spain. It has been provided by Canal Sur Televisión and it has been collected from (TNS Audiencia de Medios, 2003). Time series represent the average share for 15 minutes blocks, so the daily series are 96 elements length.

We are going to use several discretization methods and will see that the results are good in all them. A variety of discretization methods can be found in the literature. From the unsupervised algorithms: equal interval width, equal frequency interval, K-means clustering or unsupervised MCC; to supervised algorithms like ChiMerge, C ADD, 1RD, D − 2 or maximum entropy. An extensive list can be found in (Kurgan and Cios, 2004).

The methods to be evaluated in this work are: 1) Equal Width Intervals or EWI , 2) Equal Frequency Intervals or EF I , 3) C AI M (Class-Attribute Interdependence Maximization) (Kurgan and Cios, 2004), 4) Ameva (González et al, 2006), 5) C U M (González and Gavilán, 2001), and 6) DT W (Sakoe and chiba, 1978). In the following step several related tasks are accomplished: i) The discretization methods are applied over the learning subset producing a set of landmarks, ii) The landmarks are used as the limits of intervals and a symbol is assigned to each one, and iii) the series are translated into symbol chains.

The series are labelled with the name of the corresponding television station. We have selected the first 32 Wednesdays of year 2003 (32x7=224 series) as the input set of series. Other 20 Wednesdays are used as work set (140 series) to be predicted.

In the Equal Width, Equal Frequency and C U M methods, the user must specify the number of intervals to be computed. As no rule for an optimal value exists, all those methods will be calculated from 2 to 9 intervals. All the methods are applied to the learning subset and a list of interval boundaries are obtained. Individual letters are assigned in alphabetical order to each interval.

The learning system evaluates3 the number of successful identifications on the test subset using the k-neighbours algorithm for each discretization method. The application of the presented methodology achieves a 95% correct identification rate for the work set series, 133 over 140. The best discretization method for this data set was Equal Frequency Interval with 3 labels. Table 1 shows the average percentage and variance for all the methods in 200 draws for 1, 3 and 5 neighbours. In Table 1 can be observed that, although the discretization methods build the intervals following different approaches, except for some anomalous case, the results are similar, that is, the kernel is very robust in front of the discretization methods.

Another question is about the influence of the number of neighbours, in the k-neighbours algorithm, in the results. Table 1 represents the average identification for all the discretization methods with the odd values of k from 1 to 19. It shows that results are not improved with higher values of k when K_λ is used.

[3] A complete study can be found in (Cuberos et al., 2004).

Table 1. Identification Average (%) and Standard Deviation in Test Subset (200 Draws) vs. Number of neighbours

				Neighbours			
				1		3	5
Methods	Labels	Average	StDev	Average	StDev	Average	StDev
CAIM	7	90.5	4.26	89.4	4.56	89.1	4.74
AMEVA	3	91.6	2.74	89.4	2.77	89.7	2.81
CUM	2	90.7	2.86	88.4	2.91	89.0	2.98
	3	85.9	4.04	85.1	4.20	86.1	3.88
	4	75.9	6.01	71.3	5.29	70.9	5.59
	5	73.2	5.41	71.0	5.42	72.3	5.52
	6	82.4	4.21	80.8	4.03	80.8	4.95
	7	83.2	3.56	80.8	3.69	80.0	4.28
	8	85.2	3.33	82.8	2.95	82.1	3.36
	9	86.4	3.15	84.9	2.60	84.6	3.13
EFI	2	91.1	2.88	90.9	2.65	90.7	2.87
	3	**95.5**	2.13	95.4	1.98	95.1	2.02
	4	88.8	3.07	87.6	3.15	87.4	3.40
	5	85.2	3.87	85.1	4.14	85.4	3.85
	6	80.2	4.11	77.6	4.71	76.4	4.90
	7	74.6	4.78	71.7	5.31	71.0	5.37
	8	75.7	4.32	71.2	4.91	70.6	5.01
	9	74.7	5.26	70.4	5.27	69.1	6.20
EWI	2	71.0	11.5	65.2	13.2	66.5	12.9
	3	46.0	8.08	36.3	8.26	35.0	8.90
	4	71.9	11.9	67.3	14.2	68.8	14.3
	5	74.9	10.7	71.0	13.0	71.9	11.9
	6	72.3	10.9	68.3	13.7	70.3	13.6
	7	85.8	7.76	84.7	8.22	85.9	8.28
	8	75.3	9.32	73.3	10.3	74.2	11.0
	9	88.1	4.90	87.4	5.76	88.0	5.27
DTW	--	80.2	3.74	78.0	4.44	76.4	4.27

With respect to the parameter λ used in the kernel, it does not significantly affect the average of correct identification. Table 2 shows that only the C AI M method is affected by the variance of λ.

Table 2. Percentage of correct identification in the Work Set for each method vs. value of λ

	λ								
	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
CAIM	0.88	0.88	0.88	0.86	0.86	0.84	0.84	0.82	0.80
AMEVA	0.88	0.88	0.88	0.88	0.86	0.88	0.88	0.88	0.87
CUM02	0.88	0.88	0.88	0.88	0.88	0.88	0.86	0.86	0.88
EFI03	0.91	0.91	0.91	0.92	0.92	0.92	0.92	0.92	0.90
EWI09	0.85	0.85	0.85	0.85	0.84	0.84	0.85	0.87	0.85

5 Conclusions and further work

A new similarity function for symbol chains has been defined, generating in some cases a kernel. This function measures similarities between words in a dictionary when a distance measure between symbols is defined.

In the near future, we will focus on the extension of this methodology to time series with multiple attributes and other kinds of data. At the same time, we will use new data sets to extend its validation. Finally, it must be mentioned that this kernel has certain implications in the type of considered similarity that will be studied in future researches. The small influence of the λ parameter in identification tasks must be argued too.

Acknowledgements

This work was partly supported by grant PAI-2006/0619 and PAI-2006/0513 awarded by the Junta de Andalusia.

References

1. **Cristianini N. and Shawe-Taylor J.** (2000). An introduction to Support Vector Machines and other kernel-based learning methods Cambridge University press 2000.
2. **Cuberos F., Ortega J.A., Velasco F. and González L.** (2003). QSI-Alternative Labelling and Noise Sensitivity. In 17 International Workshop on Qualitative Reasoning.
3. **Cuberos F., Ortega J.A., Velasco F. and González L.** (2004). A methodology for qualitative learning in time series. In 18 International Workshop on Qualitative Reasoning...
4. **González L. and Gavilán J.** (2001). Una metodología para la construcción de histogramas.. Aplicación a los ingresos de los hogares andaluces.. XIV Reunión ASEPELT-SPAIN.
5. **González L., Velasco F., Angulo C., Ortega J.A. and Ruiz F.** (2004). Sobre núcleos, distancias y similitudes entre intervalos. Inteligencia Artificial,23, pp 111-117.
6. **González L., Velasco F., Cuberos F. and Ortega J.A.** (2006).Ameva: A discretization algorithm. Machine Learning. In revision.
7. **Kurgan L. and Cios K.** (2004). Caim discretization algorithm. IEEE transactions on Knowledge and Data Engineering, 16(2), pp. 145-153.
8. **Macskassy A., Hirsh H., Banerjee A, and Dayanik A.** (2003). Converting numerical classification into text classification. Artificial Intelligence , 143, pp. 51-77.
9. **Sakoe H. and Chiba S.** (1978). Dynamic programming algorithm optimisation for spoken word recognition. IEEE trans. On Acoustics, Speed and signal Proc.ASSP(26).
10. **TNS Audiencia de Medios** (2003). A service of Sofres AM company. www.sofresa.com

Luis González Abril *is an Associate Professor in the Dep. Of Applied Economy I at the University of Seville (Spain). He obtained his graduate in Mathematics in 1986 from the University of Sevilla and his Ph. D. degree in Economy in 2002 from the University of Seville. His researches are: machine learning and bifurcations of dynamical systems applied to economic problems.*

Francisco Velasco Morente *is a Lecturer in the Dep. Of Applied Economy I at the University of Seville (Spain). He obtained his graduate in Mathematics in 1979 from the University of Sevilla and his Ph. D. degree in Mathematics in 1991 from the University of Seville. His researches are: bifurcations of continuous and discrete dynamical systems and optimal control problems, both applied to economic problems.*

***Juan Antonio Ortega** was born in 1968 and he obtained the Ph.D. degree in Computer Science in 2000 at the Seville University in Spain. He is professor since 1992 in the Department of Languages and Computer Systems at the Seville University. His research interests are: the temporal series and the global information systems and specifically the domotic and assistencial systems. He is Head of the Centre of Computer Scientific in Andalusia (Spain).*

***Francisco Javier Cuberos** received his bachelor degree in Computer Science in 1993 and the Ph.D. degree in Computer Science in 2005 at the University of Seville (Spain). Works as Systems Administrator for Radio y Televisión de Andalucía since 1991. His main research interests are the analysis of temporal series of dynamical systems.*