A methodology for qualitative learning in time series

F.J. Cuberos (fjcuberos@rtva.es)
Dept. de Planificación. Radio Televisión de Andalucía.
José Gálvez s/n. 41092 Seville SPAIN

Juan Antonio Ortega (ortega@lsi.us.es)
Dept. de Lenguajes y Sistemas Informáticos, University of Seville.
Av. Reina Mercedes s/n. 41012 Seville SPAIN

Luis González & Francisco Velasco ({luisgon,velasco}@us.es)
Dept. de Economía Aplicada I, University of Seville.
Ramón y Cajal 1. 41018 Sevilla SPAIN

Abstract

We present an off-line methodology for the identification of series.

Given a learning set, an evaluation of the capacity of several alternatives to carry out correct identification is presented. For this, the series are transformed into symbol chains by means of several discretization methods. This transformation is done over typified and differenced series, translating the quantitative data to a qualitative description of the series evolution.

Afterwards, a distance based on a kernel between literals is used to calculate the similarity between series, and a k-neighbours algorithm is used to identify the class it belongs to.

In the interval distance defined the similarity between symbols depends on the size and position of the intervals assigned to each symbol.

The methodology has been tested with a television shares dataset presenting a high success identification ratio and it only need a neighbour to find the correct class. These characteristics are low influenced by size of the learning set.

Introduction

The study of the temporal evolution of systems is an incipient research area. It is necessary the development of new methodologies to analyze and to process the time series obtained from the evolution of these systems.

The time series, produced by a variety of applications, are usually stored in databases. It is necessary to develop new algorithms and techniques for its study.

A time series is a sequence of real values, each one representing the value of a magnitude at a point of time. A possible field of application is the comparison of time series in numeric databases. We are interested in databases obtained from the evolution of dynamic systems. A methodology to simulate semiqualitative dynamic systems it was proposed in [Ortega *et al.* (1999)].

When we are working with time-series databases, one of the biggest problems is to calculate the similarity between two given time series. The interest of a similarity measure is multiple. In this paper, this interest is focused on finding the different behaviour patterns of the system stored in a database, looking for a particular pattern, reducing the number of relevance series before applying analysis algorithms, etc, as was presented in [Cuberos *et al.* (2002)].

Many approaches have been proposed, since [Agrawal *et al.* (1993)], to solve the problem of an efficient comparison. In this paper, we propose to carry out this comparison from a qualitative perspective, taking into account the variations of the time series values. The idea of our proposal is to abstract the numerical values of the time series and to concentrate the comparison on the shape of the time series.

This work is related to previous works in similarity of temporal series, a general review was presented in [Cuberos *et al.* (2002)], and with the works in discretization of continuous attributes.

Discretization is a process of transforming a continuous attribute values into a finite number of intervals and associating with each interval a discrete value. In [Macskassy *et al.* (2003)] was shown than even on purely numerical-valued data the results of 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.

In this work some previous works are extended to define a methodology for the identification, after a learning process, of temporal series.

The rest of this paper is structured as follows: first an overview of the methodology is presented, followed by a deep review of every step involved. Next a presentation of the distance based on a kernel over literals is included, and finally a practical implementation is described. Lastly, the conclusions and ideas for future works are enumerated.

Proposed Methodology

The off-line system that implements the present methodology will be able to identify, after the study of a learning set, the new series of a working set as belonging to certain classes.

An overall diagram is presented in figure 1, some operations and processes being omitted for clarity.

Let B be a labelled database of temporal series. In the database, series from ℓ different classes are included. The series can be obtained by means of: recording the values of a magnitude (physical, biological, economical, statistical, etc) in a real system, or from a model simulation.

Each series belongs to the class represented by its label. The labels are assigned taking into account the origin of the series or by a previos expert labelling process.

A normalization process over the original set of series is applied. This process allows the comparison of series with different scales. From the possible normalization methods the methodology implements a typification. After that, a new set of series, the difference series, are obtained from the typified series.

The typified (and differenced) set of series is splitted into a learning subset and a test subset. The elements compounding both subsets are selected randomly. This splitting process takes into account the classes stratification in the database.

The next step translates the series into symbol chains. This task will be performed by applying discretization methods. As there is no universal optimal method we try several methods. Then we select the local optimal for the actual dataset.

The equal amplitude intervals, equal frequency intervals, CUM, $CAIM$ and DAC methods will be used. The first two are usually unsupervised discretization methods. The future objective of defining the methodology from an unsupervised perspective is the reason for the selection of these methods.

Especially, the $CAIM$ method has been selected because its good results and is contrasted with other methods.

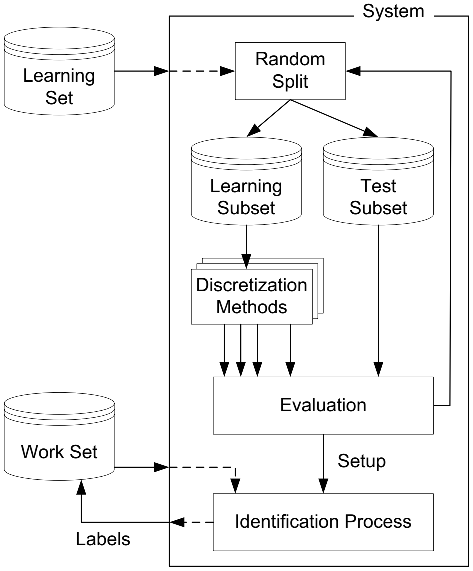


Figure 1: Proposed Methodology

The system evaluates the number of success identifications of the test subset using the k-neighbour algorithm for each discretization method. The similarity of two series is computed by an interval distance [González *et al.* (2004)].

As the original database splitting process was implemented as a random selection, all the processes described must be repeated several times. So the system eliminates the possibility of pathological combinations.

Finally, the system selects the discretization methods that computes the average better success identification ratio. This method will be used for the identification of the new series presented to the system.

In the next sections all this steps will be deeply described.

Typification

Before any other process was done with the series, a typification task is accomplished. The typification step produces a new set of series.

Let $X = \{x_0, ..., x_n\}$ be a time series, and let $X_T = \{\tilde{x_0}, ..., \tilde{x_n}\}$ be the typified temporal series obtained from X.

The series obtained with a typification process are characterized by:

- The series are unit less.
- The average is 0.
- The standard deviation is 1.
- They are invariant against scale and offset shifting, when the offset is positive, following the similarity definition presented in [Goldin and Kanellakis (1995)].

The typification is very robust to outliers in the series values produced by noise, the opposite of what happens in the normalization ised in [Cuberos *et al.* (2003)].

Let $X_D = \langle d_0, ..., d_{f-1} \rangle$ be the series of differences obtained from X_T as follows:

$$d_i = \tilde{x_i} - \tilde{x}_{i-1} \tag{1}$$

The difference series only show the evolution of the time series, so we focus on the overall shape and not on particular values.

This difference series will be used in the labelling step to produce the string of characters corresponding to X.

Figure 2 shows an example of a partial typified curve with their derivative values and the assigned label to each transition between adjacent values. The example uses a symmetrical discretization with 5 ranges whose boundaries are shown as horizontal lines.

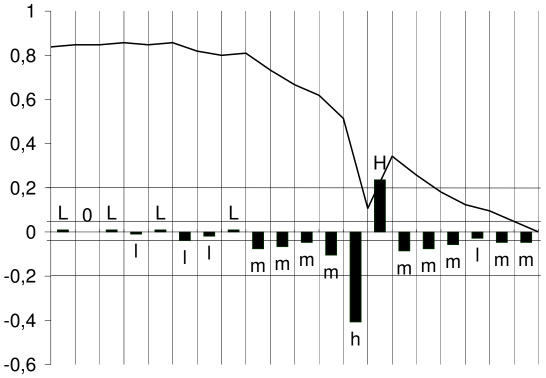


Figure 2: Sample of translation. The original time series, the values of differences (in bars) and the assigned label

Splitting Database

Now, with the set of typified series, two subsets will be created; a learning subset and a test subset. The election of the components of each set will be randomly done.

The number of series representing each different class can be variable, so the selection of elements of both subsets must take into account the stratification of classes in the database.

In this work the splitting process will follow an usual $70 - 30$ ratio for the learning and test subsets.

The splitting process is aleatory, and all the following processes are based on its output, so the final result can be affected by a pathological draw. As a preventive measure all the processes will be repeated several times. The number of iterations is a function of the data nature, but without any other study we have selected a value of 200 iterations. It is important to remember that the methodology is defined as off-line.

With the two subsets defined, the learning task can begin.

Discretization methods application

In this step several related tasks are accomplished:

- The discretization methods are applied over the learning subset producing a set of landmarks.
- The landmarks are used as the limits of intervals and a qualitative symbol is assigned to each.
- Finally the series are translated into symbol chains.

There is no universal method that computes an optimal discretization of a continuous attribute. Our approach will evaluate several methods simultaneously.

We can find a variety of discretization methods in the literature; from the unsupervised algorithms (that discretize attributes without taking into account respective class labels) as equal interval width, equal frequency interval, k-means clustering or Unsupervised MCC, to supervised algorithms like $ChiMerge$, $CADD$, $1RD$, $D-2$ or maximum entropy. An extensive list can be found in [Kurgan and Cios (2004)] and [Dougherty *et al.* (1995)].

The methods we will evaluate in this work are:

- $Equal\,Width\,Intervals$ or EWI.This is the simplest discretization method. The range of values for a continuous variable is divided into k equal size intervals. The experience shows that the division of a group of values into ranges, or intervals, with the same amplitude is the least noise sensitivity division, but it is the most losing information method as was shown in [Cuberos *et al.* (2003b)]
- $Equal\,Frequency\,Intervals$ or EFI. This method finds a set of intervals that present an approximate equal number of values. So every symbol has the same representation power in the set of series. The ends of the intervals are selected as the corresponding percentiles.
- $CAIM$. $CAIM$ (class-attribute interdependence maximization) is a supervised discretization method and it obtained good results, in terms of number of intervals, when compared with other five state-of-the-art algorithms, in [Kurgan and Cios (2004)]. The comparison included equal width and equal frequency.
- DAC method. The Discretization based on the Association Coefficient, or DAC, is a supervised discretization method defined analogously to $CAIM$. The method is based on χ^2 Test, so it has a statistical foundation. This method was defined in [González *et al.* (2004b)].
- CUM method. This method was developed in [Cochran (1977)] and implemented in [González and Gavilán (2000)]. This method makes a clustering of the initial values minimizing the average of the deviations, with the constraint that all the class marks be equally representative. This process is defined based on the statistical sampling techniques and a complete study can be found in [Cochran (1977)] and [González and Gavilán (2000)].

In the Equal Width, Equal Frequency and CUM, the user must specify the number of intervals to be computed. As there is no rule for an optimal value all those methods will be calculated from 2 to 9 intervals. We are interested in a low number of discretization intervals.

All the applications of the methods, a total of 26, are applied to the learning subset and sets of interval boundaries are obtained. A symbol, actually a single character, in alphabetical order is assigned to each interval. Each symbols is understood as a qualitative label denoting the series evolution.

This relation between intervals and characters is the key to transform the differences series generated in the typification process into strings of characters.

In previous works we defined the similarity as the number of ordered symbols in two series. Now we will use a new distance, a kernel over symbols from a discretization process.

This novel distance will be presented in the next section.

Interval Kernel

This section follows the work in [González *et al.* (2004)]. In essence, the goal in the construction of kernel functions is to guarantee the existence of an application ϕ defined from the working set, $\mathcal{X}$ (which not necessarily is provided from a previous mathematical structure) to a vectorial space equipped with a dot product named feature space, $\mathcal{F}$.

From this function ϕ, in general non linear, 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[1],

$$k(\cdot,\cdot)=\langle\phi(\cdot),\phi(\cdot)\rangle_{\mathcal{F}}$$

The kernel function $k(\cdot,\cdot)$, called *Mercel Kernel*, let us establish similarities between the original elements from their transformed ones, so a distnace between the points of origin can occasionally be defined. ϕ application, therefore, mus be able to highlight the essential characteristics of the initial set elements, so they must be considered when elaborating a similarity and distance measure. Therefore, the image space of ϕ application is known as feature space or space of characteristics.

Following the interval research approach, it will be denoted by $\mathcal{I}$ the family of all the open intervals (a,b) contained in the real line.[2] of finite dimension,,

$$\mathcal{I}=\{(a,b)\subset I\!R\,:\,a<b,a\neq-\infty,b\neq+\infty\}$$

It better denotes the intervals in the form $I=(c-r,c+r)$ where c is the center and r the radius. Thus a function ϕ is defined:

$$\phi:\mathcal{I}\rightarrow I\!R^2$$

$$\phi(I)=A\begin{pmatrix}c\\r\end{pmatrix}=\begin{pmatrix}a_{11}&a_{12}\\a_{21}&a_{22}\end{pmatrix}\begin{pmatrix}c\\r\end{pmatrix}$$

[1] $\langle\cdot,\cdot\rangle$ is denoted a dot product.

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

thus, the kernel is:

$$k(I_1, I_2) = \begin{pmatrix} c_1 & r_1 \end{pmatrix} S \begin{pmatrix} c_2 \\ r_2 \end{pmatrix}$$

and, a distance among intervals is:

$$d_1^2(I_1, I_2) = \begin{pmatrix} \Delta c & \Delta r \end{pmatrix} S \begin{pmatrix} \Delta c \\ \Delta r \end{pmatrix}$$

where $I_1 = (c_1 - r_1, c_1 + r_1)$, $I_2 = (c_2 - r_2, c_2 + r_2)$, $\Delta c = c_2 - c_1$ and $\Delta r = r_2 - r_1$. Too, A must be a non singular matrix, so ϕ be an inyective application, and $S = A^t A$ a symmetrical and positive defined matrix.

This way, the weight to give to the position of the intervals, c, ant to the size, r, can be controlled.

Thus, the conversion of a continuous attribute in labels from the construction of different class intervals allows us to use as the distance between labels the distance between intervals, as shown in the example section. In the subsequent, we will consider that symbols are letters.

Kernel over letters from disretization process

Let be an alphabet of ℓ letters which we denote:

$$\mathcal{A} = \{A_1, A_2, \cdots, A_\ell\}$$

and let $\mathcal{P}$ be a set of the all possible words with this alphabet. Lets $P1$ and $P2$ be two words on $\mathcal{P}$ that we denote:

$$P1 = P1_1 P1_2 \cdots P1_n \qquad P2 = P2_1 P2_2 \cdots P2_m$$

with $n \geq m$, $P1_i,\ P2_j \in \mathcal{A}$. A kernel is defined:

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

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

Note 1 *If the words are the same size n, then:*

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

Note 2 *This kernel is a radial basis function (R.B.F.) since it is defined like a function of a distance, $f(d(P1, P2))$.*

Property 1: *If $0 < \lambda_1 < \lambda_2 < 1$ then $K_{\lambda_1}(P1, P2) \leq K_{\lambda_2}(P1, P2)$ for all $P1,\ P2 \in \mathcal{P}$.*

Property 2: *$K_\lambda(P1, P2) \leq m$ for all $P1,\ P2 \in \mathcal{P}$ and $0 < \lambda < 1$. This threshold is hold*[3].

Property 3: *Let be $r = \max_{ij} d(A_i, A_j)$ with $A_i, A_j \in \mathcal{A}$. Then $m\lambda^{r^2} \leq K_\lambda(P1, P2)$ for all $P1,\ P2 \in \mathcal{P}$ and $0 < \lambda < 1$. This threshold is reached*[4].

Thereby, for all $0 < \lambda < 1$:

$$m\,\lambda^{r^2} \leq K_\lambda(P1, P2) \leq m, \quad \forall P1,\ P2 \in \mathcal{P}$$

Property 4 *Let $\mathcal{A}$ be an alphabet and $\mathcal{P} = \{P_1 P_2 \cdots P_n,\ P_i \in \mathcal{A}\}$ (the set of all words that have same size n). Then*

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

is a Mercer Kernel.

It is very important to know that in this language obtained from the labelling process, 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 interpret meaning from them. This kernel considers the following ones:

- The order of the letters in each words.
- The size of the words.
- Comparison letter by letter.

The λ parameter models the importance given to matching symbols versus the comparison of different symbols. For coincident symbols the value is always 1. All our tests show that, for identification purposes, the value of λ has low or none influence.

Evaluation and Identification

In this section the quality of every proposed method is evaluated. We define the quality of a discretization method as its ability identifying correctly the class to which new series from the work set belongs.

The test will try to identify every verification series by the nearest neighbour algorithm. The label of the learning series more similar to the new series is checked against the label in this series, testing if the system chooses the right label.

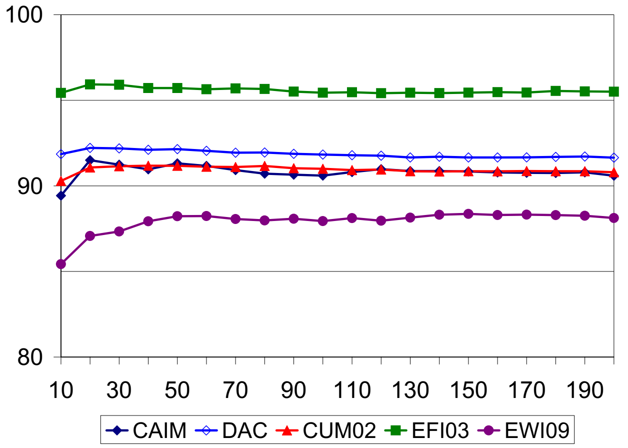


Figure 3: Identification Average (%) in Test Subset vs. Number of Draws

[3] If $P2 = P1_1 P1_2 \cdots P1_m$ then $K_\lambda(P1, P2) = m$.

[4] Let be $A = A_i$ and $B = A_j$ such that $d(A, B) = r^2$ then if $P1 = AA \cdots A$ and $P2 = BB \cdots B$ with size of $P1$, n, and size of $P2$, m. Then is true that $K_\lambda(P1, P2) = m\lambda^{r^2}$.

For the previous test we take all the series and every discretization method is applied. The application of the methods will consist in the translation of the series in chains of symbols and the calculation of the similarity between every pair by means of the interval distance.

Once all the results are obtained for each method in every try, changing the learning and test subsets, the best method for the actual dataset is elected.

After the election of one of the discretization methods, this is applied to all the series in the learning set obtaining the final set of interval boundaries.

Finally, after calculating the set of intervals produced by the best discretization method, the system that implements this methodology notifies the user the end of the learning. Now the user can present a set of new unlabelled series, the work set, and obtains an answer from the system. The answer is the class corresponding to each series using the series of the learning set as class representant.

Test

We will work with a set of television shares from the seven main television stations in Andalusia. The data has been provided by Canal Sur Television, a company of Grupo Radio Televisión de Andalucía, and generated by [Sofres].

The series represent the average share for 15 minute blocks, so the series are 96 elements length.

We have selected the first 32 Wednesdays of year 2003 as the input set of the series. Other 20 Wednesdays are used as work set. The series are labelled with the name of the corresponding television station.

		Neigbours					
		1		3		5	
Method	Labels	Avg.	StDev.	Avg.	StDev.	Avg.	StDev.
CAIM	7	90,6	4,3	89,4	4,6	89,1	4,7
DAC	3	91,7	2,7	89,4	2,8	89,8	2,8
CUM	2	90,8	2,9	88,4	2,9	89,1	3,0
	3	85,9	4,0	85,1	4,2	86,2	3,9
	4	76,0	6,0	71,3	5,3	71,0	5,6
	5	73,2	5,4	71,0	5,4	72,3	5,5
	6	82,5	4,2	80,9	4,0	80,8	5,0
	7	83,2	3,6	80,0	3,7	80,1	4,3
	8	85,3	3,3	82,8	3,0	82,1	3,4
	9	86,5	3,2	84,9	2,6	84,7	3,1
EFI	2	91,2	2,9	90,9	2,7	90,8	2,9
	3	95,5	2,1	95,4	2,0	95,2	2,0
	4	88,9	3,1	87,6	3,2	87,4	3,4
	5	85,2	3,9	85,2	4,1	85,4	3,9
	6	80,3	4,1	77,7	4,7	76,4	4,9
	7	74,7	4,8	71,8	5,3	71,1	5,4
	8	75,8	4,3	71,2	4,9	70,7	5,0
	9	74,7	5,3	70,4	5,3	69,1	6,2
EWI	2	71,0	11,5	65,3	13,2	66,6	13,0
	3	46,0	8,1	36,4	8,3	35,1	8,9
	4	71,9	12,0	67,4	14,2	68,9	14,4
	5	74,9	10,7	71,0	13,0	72,0	11,9
	6	72,4	11,0	68,4	13,7	70,4	13,6
	7	85,8	7,8	84,8	8,2	86,0	8,3
	8	75,3	9,3	73,4	10,4	74,3	11,0
	9	88,1	4,9	87,5	5,8	88,1	5,3
DTW	-	80,3	3,7	78,1	4,4	76,5	4,3

Figure 4: Identification Average (%) and Standard Deviation in Test Subset (200 Draws) vs. Number of neighbours

From the 224 series in the learning set ($32 * 7$), the learning and test subsets are completed with 154 and 70 series respectively.

The application of the presented methodology achieves a 95% correct identification rate from the work set series, 133 over 140. The best discretization method for this data set is $Equal\ Frequency\ Interval$ with 3 labels.

That level of right identification is very high but it is possible to ask about the influence of the different parameters presented.

The first open question in the proposed methodology is the number of iterations of the draw-learn-test cycle. Obviously, this value depends on the data nature.

In figure 3 we present the relation between the number of iterations and the average of correct identification in the test subset. Except for the EWI with 9 labels, the values are very stable with more than 20 iterations. Even considering EWI there is no important variation after 60 draws.

In the future perhaps a detailed study of the identifications can be used as a stop criteria for the selection of the best discretization method.

	Discretization Methods				
Number of neighbours	CAIM	DAC	CUM02	EFI03	EWI09
1	90,6	91,7	90,8	95,5	88,1
3	89,4	89,4	88,4	95,4	87,5
5	89,1	89,8	89,1	95,2	88,1
7	89,6	90,5	90,0	95,5	89,3
9	89,4	90,7	89,9	95,2	89,9
11	89,5	90,3	89,4	94,6	90,1
13	89,6	90,2	89,2	94,2	90,4
15	89,5	89,7	88,6	94,1	90,7
17	89,1	89,0	87,8	94,0	90,6
19	89,0	88,0	87,0	93,8	90,7

Figure 5: Identification Average (%) in Test Subset vs. Number of neighbours

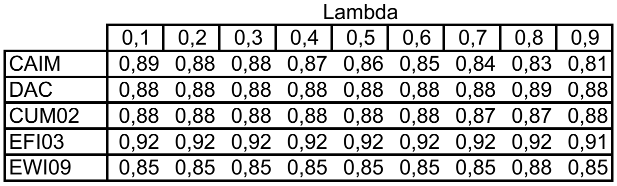
But if the average of correct identifications it is important, also the variance in the percentage of identifications can be considered to evaluate the best method. Figure 4 shows the average percentage and variance for all methods in 200 draws for 1,3 and 5 neighbours.

In this figure the identifications with *Dynamic Time Warping*, DTW, are shown. DTW [Sakoe and Chiba (1978)] is a well known method in time series community.

Another question is if a different number of neighbours in the k-neighbours algorithm has influence in the results. Although the value of k in the k-neighbours algorithm has little impact in the execution time, figure 5 shows that are not obtain better results with higher values of k. The figure represents the average identification for all the discretization methods with the odd values of k from 1 to 19.

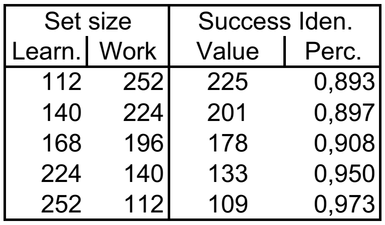
As was said in the kernel definition the value of the λ parameters has no effect in identification task. The figure 6 shows how only the $CAIM$ method is affected by the value of lambda.

Finally we want to analyze the influence of the input set of series on the total series can be also raised. We have carried



	Lambda								
	0,1	0,2	0,3	0,4	0,5	0,6	0,7	0,8	0,9
CAIM	0,89	0,88	0,88	0,87	0,86	0,85	0,84	0,83	0,81
DAC	0,88	0,88	0,88	0,88	0,88	0,88	0,88	0,89	0,88
CUM02	0,88	0,88	0,88	0,88	0,88	0,88	0,87	0,87	0,88
EFI03	0,92	0,92	0,92	0,92	0,92	0,92	0,92	0,92	0,91
EWI09	0,85	0,85	0,85	0,85	0,85	0,85	0,85	0,88	0,85

Figure 6: Percentage of correct identifications in Work Set for each method vs. value of lambda



Set size		Success Iden.	
Learn.	Work	Value	Perc.
112	252	225	0,893
140	224	201	0,897
168	196	178	0,908
224	140	133	0,950
252	112	109	0,973

Figure 7: Number of series in Learning-Work Sets vs. Identification Success in Work Set (absolute and percentage) with $EWI03$

out all the combinations of 4 day groups (as the minimal unit), and found that in the worst case the result are similar.

In figure 7, the success identifications of the work series contrasted to the relation of the sizes of the input and work sets. Following the common sense, the percentage of correct identifications is affected by the relative size of learning set.

Conclusions and Future Work

An off-line methodology has been presented which allows the identification of the class of temporal series from a set given for its learning.

A comparison is made from the series evolutions and not from the concrete values. An abstraction of the information is carried out. A supervised discretization on these evolutions is carried out, which leads to an improvement of the results.

A new distance based on an interval kernel has been defined.

In the future, our works will focus on the extension of the methodology to series with multiple attributes. At the same time, we will use new data sets to extend its validation.

The presented definition could be improved in its execution time in future works.

Finally, we must mention that this kernel has certain implications in the type of considered similarity that will be studied in future investigations. The low influence of the lambda parameter in identification tasks must be argued too.

Acknowledgments

This work was partially supported by the Spanish Interministerial Committee of Science and Technology by means of the programs DPI2001-4404-E. Moreover, it has been partly supported by the grant ACC-265-TIC-2001 and ACPAI-2003/014 given by the Junta de Andalucía.

References

[Agrawal *et al.* (1993)] *Agrawal R., Faloutsos C. y Swami A.* Efficient similarity search in sequence databases. *In Proc. of the Fourth Intl. Conf. on Foundations of Data Organization and Algorithms (FODO '93)*. Chicago.

[Cochran (1977)] *Cochran W.G.* Técnicas de muestreo. *Edit. Continental*, Mexico, 6 edition (in spanish).

[Cuberos *et al.* (2002)] *Cuberos F.J., Ortega J.A., Gasca R.M. and Toro M.* QSI - Qualitative Similarity Index. *QR-2002*. Sitges, Barcelona (Spain), pp. 45-51.

[Cuberos *et al.* (2003)] *Cuberos F.J., Ortega J.A., Velasco F. and González L.* Qsi - alternative labelling and noise sensitivity, *17º International Workshop on Qualitative Reasoning*. Brasil.

[Cuberos *et al.* (2003b)] *Cuberos F.J., Ortega J.A., Velasco F. and González L.* Qsi - Labelling and Noise Sensitivity, *CAEPIA-2004*. San Sebastián (Spain), number X, pp.445-448.

[Dougherty *et al.* (1995)] *Dougherty J., Kohavi R. and Sahami M.* Superised and Unsupervised Discretization of continuous Features, *Proceedings of the* 12^{th} *International Conference on Machine learning*, pp. 194-202, 1995.

[Goldin and Kanellakis (1995)] Goldin D.Q. and Kanellakis P.C., On similarity queries for time-series data: constraint specification and implementation. *In 1st Intl. Conf. on the Principles and Practive of Constraint Prog.* Minneapolis, 1994, pages 419-429.

[González and Gavilán (2000)] *González L. and Gavilan J.M.* Una metodología para la construcción de histogramas. Aplicación a los ingresos de los hogares andaluces, *XIV Reunión ASEPELT España* Oviedo (Spain) (in spanish).

[González *et al.* (2004)] *González L., Angulo C., Velasco F. and Ortega J.A.*, Núcleos, distancias y similitudes entre intervalos e hipercubos. To appear in *Inteligencia Artificial* (in spanish).

[González *et al.* (2004b)] *González L., Cuberos F.J., Velasco F. and Ortega J.A.*, Método de Discretización basado en el Coeficiente de Asociación. *Technical Report 005, Department of Applied Economy I,* University of Sevile (Spain) (in spanish).

[Kurgan and Cios (2004)] *Kurgan L. and Cios K.J.*, CAIM Discretization Algorithm,*IEEE Transactions on Knowledge and Data Engineering*, vol.16, no.2, pp.145-153.

[Macskassy *et al.* (2003)] *Macskassy S.A., Hirsh H., Banerjee A. and Dayanik A.A.* Converting Numerical Classification into Text Classification. *Artificial Intelligent Journal,* vol. 143, pp. 51-77.

[Ortega *et al.* (1999)] *Ortega J.A., Gasca R.M. and Toro M.* A semiqualitative methodology for reasoning about dynamic systems. 13^{th} *International Workshop on Qualitative Reasoning*. Loch Awe (Scotland), pp.169-177.

[Sakoe and Chiba (1978)] *Sakoe H. and Chiba S.* Dynamic Programmin algorithm optimization for spoken word recognition.*IEEE Trans. Acoustics, Speech and Signal Proc.*, Vol. ASSP-26.

[Sofres] *Sofres - TNS Audiencia de Medios.* A service of Sofres AM company. www.sofresam.com.