Expert Systems with Applications 36 (2009) 9924–9927

Contents lists available at ScienceDirect

Expert Systems with Applications

journal homepage: www.elsevier.com/locate/eswa

A new approach to qualitative learning in time series

L. Gonzalez-Abril [a,*], F. Velasco [a], J.A. Ortega [b], F.J. Cuberos [c]

[a] Applied Economics I Dept., Seville University, 41018 Seville, Spain
[b] Computer Languages and Systems Dept., Seville University, 41012 Seville, Spain
[c] Planning Dept. – R.T.V.A., Jose Galvez s/n, 41092 Seville, Spain

ARTICLE INFO

Keywords:
Discretization
k-nearest-neighbours
Kernel
Similarity

ABSTRACT

In this paper the *k*-nearest-neighbours (KNN) based method is presented for the classification of time series which use qualitative learning to identify similarities using kernels. To this end, time series are transformed into symbol strings by means of several discretization methods and a distance based on a kernel between symbols in ordinal scale is used to calculate the similarity between time series. Hence, the idea proposed is the consideration of the simultaneous use of symbolic representation together with a kernel based approach for classification of time series. The methodology has been tested and compared with quantitative learning from a television-viewing shared data set and has yielded a high success identification ratio.

1. Introduction

The study of the temporal tendency of systems is an incipient research area and the development of new methodologies is necessary in order to analyze and process the time series obtained for these systems. Usually these time series are stored in large databases, and a method for the simplification of information, such as a qualitative method, is therefore useful. The symbolic representation of experimental data offers a potentially powerful tool for studying dynamic behaviour and for model fitting (Lesher, Guan, & Cohen, 2000). Nevertheless, when large data sets are aggregated into a smaller data size, data tables of a higher complexity are required e.g. those of interval type instead of standard (Do & Poulet, 2005).

Given two time series, one of the major problems in database series is to obtain a similarity measure that provides an adequate approach to the problem studied. Many approaches to a similarity measure exist based on the Euclidean distance (Sivaramakrishnan & Bhattacharyya, 2004), and dynamic time warping (DTW; Sakoe & Chiba, 1978). Nevertheless, kernels methods provide a greater variety of measures than these since, for example, the Euclidean distance and L^p distances are particular cases of distances based on kernels (Schölkopf & Smola, 2002).

Ever since Agrawal, Faloutsos, and Swami (1993), many approaches have been proposed to solve the problem of an efficient comparison between time series. In this paper, we propose carrying out this comparison from a qualitative perspective, which takes into account the variations of the time series values. The idea of our proposal is to abstract the numerical values of the time series, as in Lesher et al. (2000), Lin, Keogh, Lonardi, and Chiu (2003), Robinson (2007) for example, by using the discretization of continuous values, and to concentrate the comparison on the shape of the time series. A similarity measure can be seen in Lin et al. (2003) which is defined on time series and is based on a qualitative perspective, although one of the most commonly used similarity measures is the longest common subsequence (LCS; Paterson & Dancik, 1994) which is a special case of DTW.

Almost all time series data sets are of very high dimensions which presents a major challenge since all non-trivial data mining and indexing algorithms degrade exponentially with dimensionality. Nevertheless, the qualitative perspective permits to use SVM and kernel based methods (Angulo, Anguita, Gonzalez-Abril, & Ortega, 2008; Schölkopf & Smola, 2002) since both the computational cost and the memory requirement are reduced when interval data is used instead of single data. Thus, massive data sets must be summarized into interval data sets (Bock & Diday, 1999). Another important reason for using a qualitative learning is that there are many areas of research where data sets are obtained by using statistical inference i.e. each data point is not a precise input since it is given as a confidence interval.

Many researchers have considered transforming real-value time series into symbolic representations since there is an enormous wealth of existing algorithms and data structures that allow the efficient manipulation of symbolic representations (Lin et al., 2003) from the text processing and bioinformatics. Furthermore, discretization itself may be viewed as a discovery of knowledge in that critical values in a continuous domain may be revealed.

The rest of this paper is structured as follows: in Section 2 an overview of the methodology is presented and a distance based

* Corresponding author. Tel.: +34 954554345; fax: +34 954551636.
E-mail address: luisgon@us.es (L. Gonzalez-Abril).

0957-4174/$ - see front matter © 2009 Elsevier Ltd. All rights reserved.
doi:10.1016/j.eswa.2009.01.066

on a kernel over symbol strings is also included. A practical implementation is described in Section 3. Conclusions and ideas for future work are enumerated in the final section.

2. Proposed methodology

Let B be a time series data set which is labelled with C different classes ($C > 1$). First, by using cross validation, B is split into two subsets: a learning and a test subset. Assuming that the learning set $X = \{x_0, \cdots, x_n\}$ is given, let $X_T = \{\tilde{x}_0, \cdots, \tilde{x}_n\}$ be the typified time series obtained from X where $\tilde{x}_i = \frac{x_i - \overline{X}}{S_X}$ and $\overline{X}$ and S_X are the mean and the standard deviation of X, respectively. Typification is chosen since the series is robust to outliers produced by noise in the series values, and it is invariant against scale and offset shifting when the offset is positive in the X original time series. Furthermore, it is well understood that it is meaningless to compare time series with different offsets and amplitudes (Keogh & Kasetty, 2002).

Let $X_D = \{d_1, \cdots, d_n\}$ be the series of differences obtained from X_T as follows: $d_i = \tilde{x}_i - \tilde{x}_{i-1}$ for $i = 1, \cdots, n$. The difference series shows the evolution of the time series, and hence the focus is on the overall shape and not on particular values.

In the next step several related tasks are accomplished: (*i*) the discretization methods are applied over the learning subset by producing a set of landmarks, (*ii*) the landmarks are used as the limits of intervals and a qualitative symbol is assigned to each interval, and (*iii*) finally, the series are translated into symbol strings.

We are going to evaluate our method with the following discretization methods[1]:

- *Equal-width intervals* or EWI. The range of values is divided into k equal-size intervals.
- *Equal-frequency intervals* or EFI. This method finds a set of intervals that present an approximately equal number of values.
- CUM method (González & Gavilán, 2001). This method makes a clustering of the initial values by minimizing the mean of the standard deviation in each interval, under the constraint that all the class marks must be equally representative.
- CAIM is a supervised discretization method and obtained good results in terms of number of intervals (Kurgan & Cios, 2004).
- AMEVA method is based on Chi-square statistics (Gonzalez-Abril, Cuberos, Velasco, & Ortega, 2009).

All representations allow the real-value data to be converted in a streaming fashion, with only an infinitesimal time and space overhead. Note that although the label of each time series is known, the unsupervised discretization methods EWI, EFI and CUM are used, since even if the methodology is defined from a supervised perspective, in the future an unsupervised approach can be desired.

All the applications of the methods are applied to the learning subset, and sets of interval landmarks are obtained. A symbol, actually a single character in alphabetical order, is assigned to each interval. Each symbol is understood as a qualitative label denoting the series evolution. This relation between intervals and characters is the key to transforming the difference series, generated in the typification process, into strings of characters the difference series is used in the labelling step to produce the string of characters corresponding to X (see Fig. 1). Hence, a distance defined over the strings is constructed.

Let $\mathscr{I} = \{(c - r, c + r) \subset \mathbb{R} : r > 0, a \in \mathbb{R}\}$ be the family of all the open intervals.[2] Hence, a function $\phi : \mathscr{I} \to \mathbb{R}^2$ is defined:

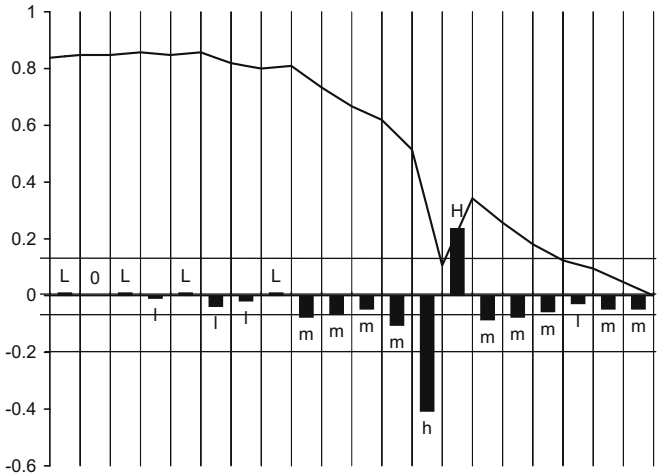

Fig. 1. Sample of translation. The original time series, the values of differences (in bars) and the assigned labels to each transition between adjacent values (**h**igh, **m**edium, **l**ow, **0**, **L**ow, **M**edium, and **H**igh). The example uses symmetrical discretization with 5 ranges whose landmarks are shown as horizontal lines.

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

From this function ϕ, a Mercer kernel (Schölkopf & Smola, 2002), denoted as $k(\cdot,\cdot)$, is defined over pairs of intervals as the inner product of their transformations, $k(\cdot,\cdot) = \langle \phi(\cdot), \phi(\cdot) \rangle_{\mathbb{R}}$. Hence a distance between intervals is defined (González, Velasco, Angulo, Ortega, & Ruiz, 2004):

$$d_1^2(I_1, I_2) = (\,\Delta c \quad \Delta r\,) S \begin{pmatrix} \Delta c \\ \Delta r \end{pmatrix} \tag{1}$$

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 $\Delta r = r_2 - r_1$. Furthermore, A must be a non-singular matrix, therefore ϕ is an injective application, and $S = A^t A$ is a symmetrical and positive defined matrix. From the A matrix, the weight given to the position of the intervals, c, and to the size, r, can be controlled.

Other kernels defined on intervals can be cited, for example, in Do and Poulet (2005) where a Gaussian kernel is defined based on the Hausdorff distance, and in Nivlet, Fournier, and Royer (2001) where interval arithmetic is used (Neumaier, 1990) from a probabilistic perspective.

The conversion of a continuous attribute into labels using the construction of different class intervals allows us to use the distance between intervals as the distance between labels. Henceforth, symbols are considered as letters since the ordinal scale can be reflected in alphabetical order.

Let $\mathscr{A} = \{A_1, A_2, \cdots, A_\ell\}$ be an alphabet of ℓ letters and let $\mathscr{P}$ be the set of all the possible words with this alphabet. Let $P1$ and $P2$ be two words on $\mathscr{P}$ that are denoted by $P1 = P1_1 P1_2 \cdots P1_n$, $P2 = P2_1 P2_2 \cdots P2_m$ with $n \geqslant m$, and $P1_i, P2_j \in \mathscr{A}$. A map is defined:

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

where $0 < \lambda < 1$ and $d(\cdot,\cdot)$ is a distance between letters. The function K_λ is a radial basis function (RBF) since is defined as a function of a distance and it is known that RBF kernels are general and efficient.

It can be verified that $m\lambda^{r^2} \leqslant K_\lambda(P1, P2) \leqslant m$, $\forall P1, P2 \in \mathscr{P}$ for all $0 < \lambda < 1$ where $r = \max_{ij} d(A_i, A_j)$, with $A_i, A_j \in \mathscr{A}$. Note that if the words are the same size n, then: $K_\lambda(P1, P2) = \sum_{i=1}^{n} \lambda^{d^2(P1_i, P2_i)}$.

The main property of this function when $n = m$ is that it is a Mercer kernel and, therefore, a distance between words can be defined as

[1] A more extensive list may be found in Kurgan and Cios (2004).

[2] By default, we are working with open intervals although it is equally possible to translate the study to closed intervals.

Table 1
Identification average (%) and standard deviation in test subset (200 draws) vs. number of neighbours.

Method	Labels	Neighbours					
		1		3		5	
		Average	Standard deviation	Average	Standard deviation	Average	Standard deviation
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	80.0	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
Euclidean distance		91,5	2,99	89,6	3,04	89,5	3,07

$$d(P1,P2)=\sqrt{2\left(n-\sum_{i=1}^{n}\lambda^{d^2(P1_i,P2_i)}\right)} \tag{2}$$

It is very important to note that $d(P1,P2)$ defines a quasi-distance between time series since two different series can define the same character string.

The λ parameter models the importance given to matching letters vs. the comparison of different symbols. For coincident letters the value is always 1.

The quality of a discretization method (its ability to correctly identify the class to which a new series from the work set belongs) can now be evaluated. The test tries to identify every verification series by using the nearest-neighbour algorithm. The label of the most similar learning series to the new series is checked against the label in this series, thereby checking 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 of transforming the series into symbol strings and calculating the similarity between every pair by means of the distance defined in (2). Once all the results are obtained for each method in every attempt, the best method for the current data set is selected by changing the learning and test subsets.

3. Experiments

The data to be considered is a set of television-viewing shares (percentage of viewers tuned into a channel at a specified time) from the seven main television channels (TV1, La2, A3, Canal Sur, Tele5, Canal+ and Canal2Andalucía) in Andalusia (Spain). The data has been provided by Canal Sur Televisión and generated by TNS Audiencia de Medios (2003).

Time series represent the average share in 15 m blocks, therefore each time series has 96 instances for a 24 h period. A single day, Wednesday, has been chosen for this study and the first 32 Wednesdays of year 2007 have been selected as the input set and the other 20 are used as the work set to be predicted.

With respect to the discretization methods EWI, EFI and CUM, the user must specify the number of intervals to be computed. As there is no rule for an optimal value, all these methods will be calculated from 2 to 9 intervals. A low number of discretization intervals is also desired since according to Catlett (1991), discretization should significantly reduce the number of possible values of the continuous attribute due to the fact that a large number of possible attribute values contributes to a slow and ineffective process of inductive machine learning.

A four fold cross validation is used and several discretization methods plus two quantitative methods are applied to the learning subset. When a discretization method is used, then a list of intervals is obtained and a letter in alphabetical order is assigned to each interval. The learning system evaluates the number of successful identifications in the test subset by using the *k*-neighbours algorithm for each method. The results are presented in Table 1 where the average percentage and variance for all the methods in 200 draws for 1, 3 and 5 neighbours are shown. Note that the execution time is not taken into account since our experiments show that it is not significant. Furthermore, it can be seen that the qualitative methods are competitive with the quantitative methods.

The application of the methodology presented achieves a 95% correct identification rate for the work set series. The best discretization method for this data set was EFI with 3 labels, the same number of labels as for AMEVA which found the optimum number of labels directly. Furthermore, it can be observed in Table 1 that although the discretization methods build the intervals by follow-

Table 2
Percentage of correct identification in the work set for each method vs. λ value.

	Lambda								
	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	0.88	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

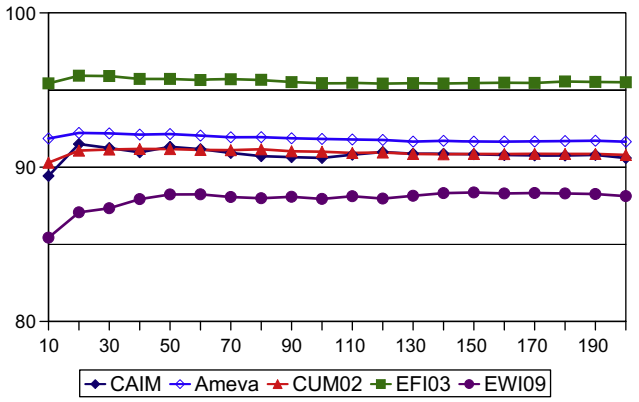

Fig. 2. Identification average (%) in test subset vs. number of draws.

ing different approaches, except for some anomalous cases, the results are similar, that is, the kernel is very robust when confronted with the discretization methods.

The level of correct identification by these methods are very high although it is possible to question the influence of the different parameters presented. Initially, the influence of the λ parameters of the kernel definition were studied. However, this had no effect on the identification task as can be seen in Table 2 where the best methods EWI, EFI and CUM are shown. Note that only the CAIM method is affected by the value of λ.

Moreover, the relation between the number of iterations and the average of correct identification in the test subset is given in Fig. 2. With the sole exception of EWI with 9 labels, the values are very stable with more than 20 iterations. Even when taking EWI into account, there is no important variation after 60 draws.

4. Conclusions and future work

An off-line methodology has been presented which allows the identification of the class of time series. A comparison between series is made based on the series tendencies and not based on specified values. A supervised discretization on these tendencies is carried out, which leads to an improvement of the results.

The novel features of the new method come from the symbolic representation of time series and the distance based on a kernel between symbols. Experiments on a television-viewing share data set are also conducted to verify the feasibility of the proposed method.

Our research is focused on finding the different behaviour patterns of the system stored in a database, looking for a particular pattern, and reducing the number to only relevant series before applying analysis algorithms, as presented in Cuberos, Ortega, Gasca, and Toro (2002). Thus this paper is a first approach to resolving this problem. In the future, our work will focus on the extension of the methodology to include series with multiple attributes. At the same time, new data sets will be used to reinforce its validation.

Finally, it should be mentioned that this kernel has certain implications in the type of similarity considered and these implications will be studied in future research. With respect to the classification methods, we are interested in applying support vector machines in line with our other research (Angulo et al., 2008; Angulo, Ruiz, González, & Ortega, 2006).

Acknowledgements

This work was partly supported by the FAMENET–InCare (TSI2006-13390-C02-02) from the Spanish Ministry of Education and Science and CUBICO (P06-TIC-02141) from Andalusian Government.

References

Agrawal, R., Faloutsos, & C., Swami, A. (1993). Efficient similarity search in sequence databases. In *Proceedings of the fourth international conference on foundations of data organization and algorithms* (FODO'93).

Angulo, C., Anguita, D., Gonzalez-Abril, L., & Ortega, J. A. (2008). Support vector machines for interval discriminant analysis. *Neurocomputing, 71*(7–9), 1220–1229.

Angulo, C., Ruiz, F., González, L., & Ortega, J. A. (2006). Multi-classification by using tri-class SVM. *Neural Proceeding Letters, 23*(1), 89–101.

Bock, H., & Diday, K. (1999). *Analysis of symbolic data*. Springer-Verlag.

Catlett, J. (1991). On changing continuous attributes into ordered discrete attributes. In *Proceedings of European working session on learning*. pp. 164–178.

Cuberos, F., Ortega, J., Gasca, R., & Toro, M. (2002). Qsi – Qualitative similarity index. In *16th International workshop on qualitative reasoning*.

Do, T. -N., & Poulet, F. (2005). Kernel-based methods and visualization for interval data mining. In *ASMDA'05, international symposium on applied stochastic models and data analysis*. pp. 345–354.

Gonzalez-Abril, L., Cuberos, F., Velasco, F., & Ortega, J. (2009). Ameva: Autonomous discretization algorithm. *Expert System with Application, 36*(3), 5327–5332.

González, L., & Gavilán, J. (2001). Una metodología para la construcción de histogramas. Aplicación a los ingresos de los hogares andaluces. In *XIV Reunión ASEPELT–Spain* (in Spanish).

González, L., Velasco, F., Angulo, C., Ortega, J., & Ruiz, F. (2004). Sobre núcleos, distancias y similitudes entre intervalos. *Inteligencia artificial. Revista Iberoamericana de IA, 8*(23), 113–119. in Spanish.

Keogh, E., & Kasetty, S. (2002). On the need for time series data mining benchmarks: A survey and empirical demonstration. In *Proceedings of the 8th ACM SIGKDD international conference on knowledge discovery and data mining*. pp. 102–111.

Kurgan, L., & Cios, K. (2004). CAIM discretization algorithm. *IEEE Transactions on Knowledge and Data Engineering, 16*(2), 145–153.

Lesher, S., Guan, L., & Cohen, A. H. (2000). Symbolic time-series analysis of neural data. *Neurocomputing, 32–33*, 1073–1081.

Lin, J., Keogh, E., Lonardi, S., & Chiu, B. (2003). A symbolic representation of time series, with implications for streaming algorithms. In *DMKD 2003: Proceedings of the 8th ACM SIGMOD workshop on Research issues in data mining and knowledge discovery* (pp. 2–11). New York, NY, USA: ACM.

Neumaier, A. (1990). *Intervals methods for systems and equations*. Cambridge University Press.

Nivlet, P., Fournier, F., & Royer, J. -J. (2001). Interval discriminant analysis: An efficient method to integrate errors in supervised pattern recognition. In *ISIPTA*. pp. 284–292.

Paterson, M., & Dancik, V. (1994). Longest common subsequences. In *Proceedings of the 19th MFCS, lecture notes in computer science* (vol. 841, pp. 127–142). Berlin: Springer.

Robinson, S. (2007). A statistical process control approach to selecting a warm-up period for a discrete-event simulation. *European Journal of Operational Research, 176*, 332–346.

Sakoe, H., & Chiba, S. (1978). Dynamic programming algorithm optimization for spoken word recognition. *IEEE Transactions on Acoustics, Speech, and Signal Processing, 26*(1), 43–49.

Schölkopf, B., & Smola, A. J. (2002). *Learning with kernels*. Cambridge, MA: The MIT Press.

Schölkopf, B., & Smola, A. J. (2002). *Learning with kernel*. MIT Press.

Sivaramakrishnan, K. R., & Bhattacharyya, C. (2004). Time series classification for online Tamil handwritten character recognition – A kernel based approach. In *ICONIP*. pp. 800–805.

TNS Audiencia de Medios. (2003). A service of sofres AM company. <www.sofresam.com>.