# Entropy Based Linear Approximation Algorithm for Time Series Discretization

Acosta-Mesa Héctor-Gabriel, Cruz-Ramírez Nicandro and García-López Daniel-Alejandro

Universidad Veracruzana, Departamento de Inteligencia Artificial, Sebastian Camacho No. 5, Col. Centro C.P. 91000, Xalapa, Veracruz, México.
{heacosta, ncruz}@uv.mx, dalexgarcia@gmail.com

**Abstract.** Time series data mining is a relatively new sub-area of data mining, in which the temporal dimension of data introduces new challenges for classification and clustering tasks. The huge amount of information contained in temporal databases requires efficient representations, not only to reduce dimensionality, but also to preserve the relevant information for efficient classification. Many approaches have been proposed to represent temporal data in a discrete form. However, most of them are oriented to data compression, rather than to information maximization. In this work we propose new time series discretization algorithm called EBLA2. The basic idea behind EBLA2 is to minimize the entropy of the temporal patterns over their class labels after finding a minimum set of intervals from which the continuous values of the temporal database can be discretized. Under a similar approach, the algorithm is able to find the minimum time series length to represent the complete time series database.

## 1 Introduction

The measurement of observations that evolve over time is becoming a very common task due to the huge amount of applications that produce this kind of data; for example, in medical or industrial applications. Unfortunately, temporal data produces enormous databases that require efficient representations, not only to reduce dimensionality, but also to preserve the relevant information for efficient classification. Moreover, most of the algorithms for classification work only with discrete data [10]. Many approaches have been proposed to represent temporal data [8][6], all of them are oriented to data compression, rather than to information maximization; that is to say, those representations transform times series of length $N$, into a set of $n$ coefficients, where $n < N$ . These data compression processes are only intended to reduce dimensionality of the data. However, they do not take in account if the new representation preserves the relevant information to maintain the membership of the observations to the label class associated to each one. Discretization algorithms which maximize information on data will improve the efficiency of classifications tasks performed on it, and also could be used as a feature detection tools [4].

*© A. Gelbukh, Á. Kuri (Eds.)*
*Advances in Artificial Intelligence and Applications*
*Research in Computer Science 32, 2007, pp. 214–224*

*Received 17/06/07*
*Accepted 31/08/07*
*Final version 21/09/07*

In most of the algorithms proposed to discretize time series, the user requires to specify a set of parameters to perform the transformation; for example, the number of segments (word size) to divide the times series length, and the number of intervals (alphabet) required to compress the time series values. Without a previous analysis of the temporal data, it is very difficult to know the proper parameter values to obtain a good discrete representation of data. However, in practice it is assumed that the parameters are known.

In the present work, we propose a new algorithm for time series database discretization called EBLA2 (Entropy Based Linear Approximation). Under this approach the data transformation is done in a supervised fashion. The goal is to find the minimum word size and alphabet size, and the range values for each interval which minimize the entropy of the discrete representation with respect on the class label of the database. The content of the paper is presented as follows: in section 2 the related work is discussed. Section 3 describes the proposal and main concepts involved in its definition. In section 4 the algorithm EBLA2 is explained in detail. Section 5 describes the experiments carried out and the data used to evaluate the performance of the algorithm. Section 6 shows the results obtained over the experiments. Section 7 presents a discussion, and finally in section 8, conclusions and future work are presented.

## 2 Related work and motivation

Based on the divergence of Kullback-Leibler, in [9] a time series discretization method is proposed, the core of the algorithm is to detect the persistent states on the time series that consist of recurring persistent states produced by an underlying process, this is a unsupervised method because does not require specification of class label. The applicability of this method is restricted to the existence of persistent states in the time series, something that is not very common in most of real applications. The persistent algorithm processes a single time series at a time, then the discretization criterion is not generalized to the complete dataset.

Dimitrova et al [2], using different approach represents time series as a multi-connected graph; under this representation, similar time series are grouped into a graph model. The algorithm focuses on minimizing the number of connected nodes (graph) to represent a model time series under different criterion based on Single-Link Clustering (SLC) algorithm, plus one criterion to consider the entropy to determine which arcs on the graph to be deleted. Each node in the graph represent one point in the time series. The main drawback of this algorithm is then its computational complexity ($O(MN^4)$,where $N$ is the number of nodes and $M$ is the number of time series contained on the database) that makes it difficult to apply on big datasets.

More recently, an algorithm called Symbolic Aggregate Approximation (SAX) has been proposed [8]. This algorithm is based the Piecewise Aggregate Approximation representation [1]. SAX algorithm requires the user to define an alphabet ($A$) with fixed interval ranges and a word size ($Ws$) as parameters. After a nor-

malization procedure, times series length is partitioned into ($Ws$) segments, the corresponding values are mapped to one of $A$ discrete values through the use of a normal probability density function (PDF). One advantage of this representation is that, once the dataset has been transformed, a smoothed version of the original data can be recovered using the PDF. Although SAX has been created for streaming data, it has proved to be an efficient representation for classification and clustering. However, as SAX works in a unsupervised fashion. it does not take advantage of class labels to improve classification performance. The present work, responds to the necessity of having an algorithm that not only discretizes temporal databases in order to reduce dimensionality, but also that automatically determines the reduction scale of the time series (word size) and the number of intervals (and its different ranges) that maximize the classification accuracy. The proposed algorithm EBLA2, takes as antecedent the discretization algorithm Class-Attribute Interdependence Maximization (CAIM) reported in [6]. CAIM algorithm works with static data in a supervised approach, the main idea of CAIM is to keep the interdependence relationship between attributes and class labels using a proposed information gain metric, called CAIM measure. In a similar way, EBLA2 proposes a metric to define an equivalent measure for time series discretization, it uses information gain measured in terms of entropy. For dimensionality reduction the algorithm takes as an antecedent the time series representation Piecewise Aggregate Approximation (PAA)[1], it consists of obtaining the mean values of a predefined segment over the time series. In the following section a complete description of the proposal is introduced.

## 3 Proposal

Discretization is concerned the process to map variables with continuous values into discrete values. This process has been widely used to compress data to facilitate computation in terms of space and time. More formally given the data domain $x|x \in \mathbb{R}$, where $\mathbb{R}$ is the set of reals and the discretization scheme(D) $D = \{[d_0, d1], (d_1, d_2], ..., (d_{n-1}, d_n]\}$ where $d_0$ and the $d_n$ are the minimum and maximum value of $x$ respectively. Every pair of values represents an interval, one of each maps the specific range of continuous values to one element of a discrete set {1..m}. Where $m$ is called the discretization degree and $d_i|i = 1..n$ are the interval limits, also know as a cut points [6]. Discretization process can be split in two main jobs. The first one is to find the number of discrete groups to do the mapping from continuous to discrete. The second one is to define the range or limits of each interval in the continuous domain [6]. Both jobs are done by EBLA2 using a principle of information gain based on entropy. The aim of the algorithm is to find the number of intervals, and their limits from which the membership of the resulting discrete models are clustered with respect to the class label. As explained in the following section, the discretization scheme is computed for the whole dataset, i.e. all the time series values are considered to find the discretization scheme with minimum entropy.

Although it is almost the same process, for the sake of clarity we have divided the time series discretization in two phases . The first is to find the alphabet size and the second is to find the word size. On each phase the objective is not only to find the number of discretization values, but also the range of the interval. It can be thought of as a discretization on both axis, $x$ and $y$ respectively, see figure 1 . EBLA2 is based on PAA representation, which consists in obtaining the mean values of each segment in which the time series is divided.

### 3.1 Dimensionality reduction

A simple and efficient time series representation is Piecewise Aggregate Approximation (PAA), where each segment has equal size and the number of segments is determined for user. Often, a priori information is not available, so the goal is to find the number of intervals with different size for reduction of dimensionality at the same time as maintaining the information with respect to the class label. Let $C$ be a time series with length $n$ represented in a $w$-dimensional space (word) as a vector $\overline{C} = \overline{c_1}, ..., \overline{c}_w$. and $T = \{t_1, t_2, ..., t_w\}$ be the discretization scheme where $t_i$ is the interval of time of $i$ segment of $\overline{C}$. Where the $i$ element of $\overline{C}$,is computed as:

$$\overline{c}_i = \frac{1}{|t_i|} \sum_{j=1}^{|t_i|} C_{t_j} \tag{1}$$

where: $|t_i|$ is the number of elements of $t_i$. Figure 1 shows a continuous time series $S_r|s_i \in \mathbb{R}, i = 1..300$ and its corresponding discrete representation $S_d|s_j, j = 1..3$ (word size =3), with $D = \{[c_0, c_1], ...(c_4, c_5]\}$ where $c_0 = -10,..,c_5 = 60$. As shown in figure 1, every single value is mapped to one out of three discrete values, according to the intervals found by the algorithm $[c_0, c_1] = 1, ..., (c_4, c_5] = 5$ (see horizontal doted lines), and intervals time $T = \{t_1, t_2, t_3\}$ where $t_1 = [1, 66]$, $t_2 = [67, 150]$, $t_3 = [151, 300]$ (see vertical doted lines)

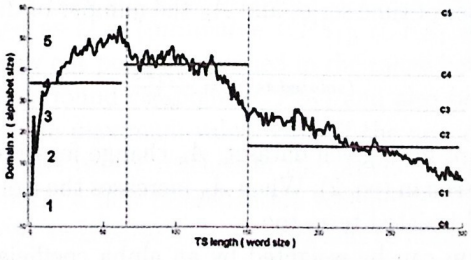


**Fig. 1.** Time series representation. Raw data $S_r$ is represented in EBLA2 as $S_d$={4,5,2}

### 3.2 Utility Measure

Most discretization algorithms require heuristics to avoid the a priori definition of the alphabet size. For example, in temporal databases, the information criterion and [2], and the persistence score [9], information entropy maximization

(IEM), information gain, maximum entropy, Petterson-Niblett and Minimum Description Length (MDL) [6],[3] has been used. In the present work, the proposed criterion to select the optimal cut points is based on the information gain measure, defined for a specific discretization scheme and its corresponding class labels. Formally, the information gain based on entropy can be stated as:

$$Gain(S, A) = Entropy(S) - \sum_{v \in A_n} \frac{\#S_v}{\#S} Entropy(S_v) \tag{2}$$

Where: $S$ and $A$ are two different set time series.
$A_n \subseteq S | a_i \in A_n \wedge a_i \notin (A_n \setminus \{a_i\}), i = 1..n$
$\#S_v$ is the number of time series with value $v$ in $S$
$\#S$ is the number of time series in $S$
The entropy of $S$ is given by:

$$Entropy(S) = \sum_{i=1}^{c} -p_i \, log_2 \, p_i \tag{3}$$

Where: $c$ is the number of classes.
$p_i$ is the probability of class $i$ in $S$
The entropy of $S$ when it takes the value $S_v$ is

$$Entropy(S_v) = \sum_{i=1}^{c} p(S|v)_i \, log_2 \, p(S|v)_i \tag{4}$$

Where: $c$ is the number of classes.
$p(S|v)_i$ is the conditional probability of class $i$ in $S$ given a time series with value $v$.

Because it is possible to find different cut points with equal information gain, we propose to aggregate one term to the selection criterion. This new term is a heuristic to select one of the tied candidates, this term biases the selection to those cut points that generates higher different time series patterns given a discretization scheme. In other words, it prefers a discretization schemes which generate more isolated temporal patterns. This metric is computed as follows: let $\#S$ be the number of time series and $A_n$ the number of different time series:

$$Isolated\_term = \frac{\frac{A_n}{log_2 \#S}}{\#S} \tag{5}$$

$\#S$ remains constant for a given dataset. $A_n$ change for different discretization scheme as it can be seen in (eq. 2). When $A_n$ increases the isolated term increases and when decreased isolated term too.

The Isolated term can be weighted by an alpha coefficient to modulate its influence over the entire weight. The behavior of the isolated term in a database with 500 instances is shown in figure 2.

The goodness of a cut point is given by the sum of terms (eq. 2) and (eq. 5)

$$Utility = Gain + Isolated\_term \tag{6}$$

It is important to remark that the entire time series is considered as one attribute, and that the discretization scheme is considered for the whole database. It allows the algorithm to find a good global solution, that minimizes the entropy on data.

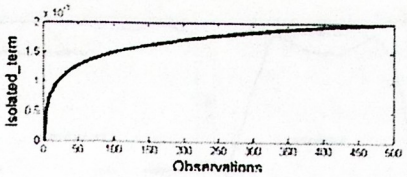


**Fig. 2.** Isolated term behavior. Cut points which discernes higher amounts of observations are better ranked

## 4 Algorithm

Under the utility measure (eq. 6) explained above, the time series discretization process can be thought of as the search on the space of all possible discretization schemes. The discretization problem on time series can be divided into two subproblems, the first one is to find the number and range of intervals of time that maintain the relevant information (alphabet), and the second is the discretization of data on each interval of time (word size). EBLA2 has been designed to solve both problems. A discretization scheme is conformed by a set of intervals defined by the cut points. In order to select each interval to be included in the discretization scheme, a set of prospectus cut points has to be created and evaluated. Some algorithms for discretization of static data, like CAIM [6], use to create the group of cut point candidates as the middle points of the different adjacent values contained in all existing continuous values (see figure 3).

To do this job, on the first iteration the algorithm searches for a cut point (between the minimum value and the maximum value contained all over the continuous range of the time series values) with maximum utility in terms of formula (6). Once the first cut point is found ($CPS_1$), the first discretization scheme is constructed as $D = \{[minvalue, CPS_1], (CPS_1, maxvalue]\}$. On the second iteration, a new cut point is searched in the range between the minimum value and the previous found cut point ($CPS_1$) and also between the previous found cut point and the maximum value. Only if the new cut point found has a bigger utility than the utility reached in the previous iteration, the new cut point is inserted to form a new discretization scheme $D = \{[minvalue, CPS_1], (CPS_1, CPS_2], (CPS_2, maxvalue]\}$. It is important to note, that on every single iteration a new set of utility values is generated. The iterative process continues until the new cut point does not produce a utility improvement. The cut point selection explained above can be extremely expensive, considering that a continuous attribute could have a huge number of different values, and that many of them could be very similar. A more efficient approach used in this work is, instead to use all the different values, to form representative values computed as percentiles. On the present algorithm we evaluate the cut points locating the percentiles defined from 0.0% to 100% on increments of 0.1%.

On the second phase of the discretization process (word size computation), the cut points represent intervals of time, and then are intended, to reduce the

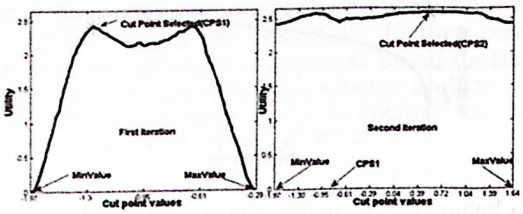


**Fig. 3.** Cuts point selection. (a) Cut points obtained in the first iteration $D = \{[minvalue, CPS_1], (CPS_1, maxvalue]\}$ (b) Discretization scheme obtained on the second iteration $D = \{[minvalue, CPS_1], (CPS_1, CPS_2], (CPS_2, maxvalue]\}$

time series length. This process consists in joining adjacent intervals keeping the information gain. The utility is computed in the same way as computed to find the alphabet, but in this case instead to add cut points , the objective is to delete cut points. The starting each point of time series is considered as a cut point and the job is to delete as much cut points as accessible keeping the classification accuracy reached on the previous phase (alphabet computation), a cut point is consider every single observation on the time series. Using an iterative approach, the entropy is computed for every deleted point. At the end of this cycle, the cut point with more information is deleted and the cycle continues until a new delete cut point does not improve the entropy. In this case point selection is made finding which prospect to eliminate from all prospectus cut points. This process can be though of as the elimination of those observation that does not affect the classification performance.

It is important to see, that this second phase has to be done after to find the best alphabet, having in mind to reduce the time series length preserving the relevant information on data. The pseudocode of the EBLA2 algorithm is shown below.

First phase:

$VD = sort(Unique(S_p))$
$C = Percentiles(VD)$
$MiddlePoints = \sum_{i=1}^{SizeC-1} \frac{C_i + C_{i+1}}{2}$
$B = min(C) \cup max(C)$
$GlobalUtility = 0$
$L_b = -inf$
While $L_b >$ $GlobalUtility$ and not empty(C)
    $L = \{\}$
    foreach c ∈ C and c ∉ B
        $L = L \cup FUtility(D(S, T, G(B \cup \{c\})))$
    $b = max_i(L)$ //Index of value maximum of L
    if $L_b > GlobalUtility$
        $B = B \cup c_b$
        $GlobalUtility = L_b$
    $C = C \setminus \{c_b\}$
return $B$

Where $S = \{s_i | s_i \in \mathbb{R}, i = 1..n\}$ represents the numeric values of a time series $S$ with length $n$. $S_p$ is the set of $p$ time series. $T = \{t_i | i = 1..r\}$ is the temporal

scheme with $r$ intervals of time is divided. $C = \{C_k \in \mathbb{R}, k = 1..m\}$ is the set of $m$ prospectus cut points. $G$ is a function for generating one discretization scheme $E$, given a set of limits $B$. $B = \{C_a | C_a \in C\}$ is a cut points set on $C$. $E = \{(B_{b-1}, B_b] | B_b > B_{b-1} b = 2..g+1\}$is a discretization scheme of length $g$. $D$ is a discretization function to transform the set $S_p$ into a discrete form given a discretization scheme. $FUtility$ is the utility value of $D$ computed as described in (eq. 6). In the first phase the process starts using a discretization scheme: The value of $T$ contains all the intervals of the complete time series.

On the first phase, the algorithm starts using a discretization scheme with a unique interval. Iteratively, the algorithm increments the number of intervals using the cut point selection procedure described above. The stop criterion was defined when the addition of a new cut point does not provide an improvement in terms of information gain. Specifically, in our implementation we used the difference of (eq. 6) at a consecutive iterations evaluated as:

$$Utility^{iteration} < Utility^{iteration-1}$$

Second phase:

$C = \{2, .., n\}$
$T = \{1, ... n\}$
$GlobalUtility = FUtility(D(S, T, G(B)))$
$L_b = inf$
While $min(L_b) >= GlobalUtility$ and not empty(C)
    L={}
    foreach c $\in$ C and c $\notin$ B
        $L = L \cup FUtility(D(S, H(T, c), G(B)))$
    end
    $b = max_i(L)$ //Index of value maximum of L
    if min($L_b$) >= $GlobalUtility$ and not empty(C)
        $T = H(T, L_b)$
return $T$

In the second phase, the algorithm joins the consecutive bits of the time series in the following way : Let $H(P, p)$ be a function that joins (with the left interval of $p$) two adjacent intervals of $p$ in $P$ . It is important to remember that the value of B (alphabet) was computed in the previous phase.

## 5 Experiments

The performance of the algorithm was tested using twenty databases available on the time series classification/clustering WEB page (www.cs.ucr.edu/~eamonn/time_series_data), see five first columns in table 1.

Different time series representations have been evaluated using these databases [?]. It is important to remark that most of these representations have been developed to compress data, rather that to improve classification performance. One of the most efficient representation recently proposed is Symbolic Aggregate Approximation (SAX)[8]. Although SAX representation has been designed to discretize time series for streaming data purposes, it has shows a good performance on classification and clustering tasks. Because as far as we know at the time at which this work was developed there was no a representation specifically designed to optimize classification performance, the performance of EBLA2 was compared to those results obtained by SAX.

**Table 1.** Databases properties, alphabet and word sizes obtained for EBLA2

| DataSet | Num. Classes | Size training | Size testing | Length TS | Word Size | Alphabet Size |
|---|---|---|---|---|---|---|
| 50Words | 50 | 450 | 455 | 270 | 40 | 2 |
| Adiac | 37 | 390 | 391 | 176 | 90 | 9 |
| Beef | 5 | 30 | 30 | 470 | 23 | 3 |
| CBF | 3 | 30 | 900 | 128 | 64 | 2 |
| Coffee | 2 | 28 | 28 | 286 | 75 | 3 |
| ECG200 | 2 | 100 | 100 | 96 | 48 | 2 |
| Fish | 7 | 175 | 175 | 463 | 119 | 3 |
| Face(All) | 14 | 560 | 1690 | 131 | 33 | 3 |
| Face(Four) | 4 | 24 | 88 | 350 | 45 | 2 |
| Gun Point | 2 | 50 | 150 | 150 | 24 | 2 |
| Lighting 2 | 2 | 60 | 61 | 637 | 80 | 2 |
| Lighting 7 | 7 | 70 | 73 | 319 | 81 | 2 |
| Osu Leaf | 6 | 200 | 242 | 427 | 54 | 2 |
| Olive Oil | 4 | 30 | 30 | 570 | 287 | 3 |
| Swedish Leaf | 15 | 500 | 625 | 128 | 67 | 5 |
| Trace | 4 | 100 | 100 | 275 | 144 | 3 |
| Two Pattern | 4 | 1000 | 4000 | 128 | 64 | 2 |
| Control Chart | 6 | 300 | 300 | 60 | 30 | 3 |
| Wafer | 2 | 1000 | 6174 | 152 | 91 | 15 |
| Yoga | 2 | 300 | 3000 | 426 | 119 | 3 |

### 5.1 Classification

One nearest neighbor (1NN) was used as a classification method. Error rate was computed using Leave-One-Out Cross Validation (LOO). The discretization scheme was obtained over the training set of data. The similarity measure used in 1-NN, to evaluate the EBLA2 algorithm was the Euclidean distance. Classification performance was also evaluated using the raw data (continuous time series). It is important to remark to SAX uses its own similarity measure [8]. SAX similarity measure is defined as: Let $\hat{Q}$ and $\hat{C}$ be the SAX discrete representation of time series $Q$ and $C$. Let $w$ be the word size. *dist* function is a specific defined similarity measure defined by SAX to decode the SAX representation [8].

$$MINDIST(\hat{Q}, \hat{C}) = \sqrt{\frac{n}{w}} \sqrt{\sum_{i=1}^{w} (dist(\hat{q}_i, \hat{c}_i))^2} \tag{7}$$

## 6 Results

Results of the data reduction alphabet and word size obtained automatically after to applying EBLA2 to 20 database are shown in the 6th and 7th column of table 1. The results obtained showed that EBLA2 representation needs smaller alphabets that SAX representation to obtain similar error rates. Figure 4 shows error rates obtained on the twenty database, on the graph is observed that most of the cases EBLA2 reached error rate smaller than those obtained by SAX using the same parameters (alphabet size and word size).

## 7 Discussion

Generally speaking, error rates reached using EBLA2 representation were smaller that those obtained by SAX tested using the same parameters (word size and alphabet size), however higher alphabet sizes on SAX obtains smaller error rates than EBLA2.

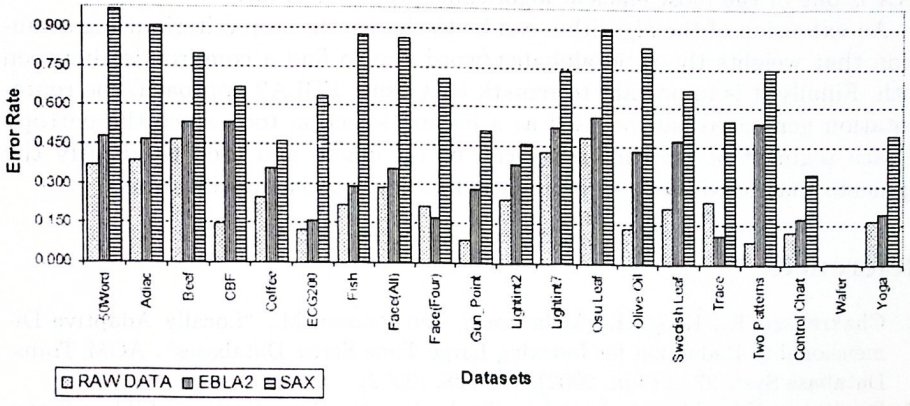


**Fig. 4.** Error rate obtain for EBLA2 and SAX using the parameters suggested by EBLA2. Error rate reached using raw data is shown as well

This seems to indicate that EBLA2 finds a local minimum derived of its greedy approach. In order to solve this, aleatory movements to be required to implement at the beginning of the search. The proposed discretization approach used by EBLA2 is an efficient technique to automatically compute the parameter for discretization(alphabet and word size), reaching competitive error rates. This characteristic makes EBLA2 very useful because some times it is very difficult to know a priori a good combination of alphabet and word size for a given dataset.

## 8 Conclusions and future work

In the present work a new algorithm for supervised discretization on time series data called EBLA2, has been proposed. Given a labeled dataset containing temporal data, EBLA2 algorithm automatically computes not only the alphabet size but also the limits which define the continuous intervals of each alphabet letter. EBLA2 also computes the number and sizes of the intervals to define the time series length. The approach proposed in this work is to evaluate different discretization schemes trying to minimize their associated entropy with respect to the class labels.

The efficiency of the algorithm was evaluated using twenty databases and compared to one of the more efficient time series discretization algorithm called

SAX and raw data. Generally, error rates reached using EBLA2 representation were smaller that those obtained by SAX tested using the same parameters (alphabet and word size). The advantage that EBLA2 does not requires a priori parameter because they are automatically calculated.

Although SAX was not developed for supervised discretization, we decided to compare EBLA2 with SAX because at the time of this publication, there was no algorithm available for supervised discretization of time series, and because SAX is one of the most efficient algorithms proposed so far.

An extension of the algorithm can be to change the stop criteria using a function that weights the ratio alphabet/word size to find a compromise between both. Finally it is important to remark that using EBLA2 approach, the representation generated can be used as a feature selection tool, since the entropy of each segment of the time series can be calculated and used to identify the segments that provides more information to develop the classification.

## References

1. Chakrabarti K., Keogh E., Mehrotra S. and Pazzani M., "Locally Adaptive Dimensionality Reduction for Indexing Large Time Series Databases". ACM Trans. Database Syst. 27, 2 (Jun. 2002), 188-228, (2002).
2. Dimitrova, E.S., McGee, J. and Laubenbacher, E.: Discretization of Time Series Data, eprint arXiv:q-bio/0505028, (2005).
3. Fayyad U. and Irani K., "Multi-Interval Discretization of Continuous-Valued Attributes for Classification Learning", In Proceedings of the 13th International Joint Conference on Artificial Intelligence, pp. 1022–1027, Chambery, France, Morgan Kaufmann, (1993).
4. Geurts P. , "Pattern Extraction for Time Series Classification", In Proceedings of PKDD 2001, 5th European Conference on Principles of Data Mining and Knowledge Discovery(September 03 - 05, 2001). L. D. Raedt and A. Siebes, Eds. Lecture Notes In Computer Science, vol. 2168. Springer-Verlag, London, 115-127, (2001).
5. Han, J. and Kamber, M.: *Data mining, Concepts and techniques* Morgan Kaufmann, (2001).
6. Kurgan L. and Cios K., "CAIM Discretization Algorithm", IEEE Transactions On Knowledge And Data Engineering, 16(2), 145-153, (2004).
7. Last M., Kandel A., Bunke H. *Data mining in time series databases*, World Scientific Pub Co Inc, (2004).
8. Lin J., Keogh E., Lonardi S. and Chiu B., "A symbolic representation of time series, with implications for streaming algorithms", In Proceedings of the 8th ACM SIGMOD Workshop on Research Issues in Data Mining and Knowledge Discovery (San Diego, California, June 13 - 13, 2003). DMKD '03. ACM Press, New York, NY, 2-11, (2003).
9. Mörchen F. and Ultsch A., "Optimizing Time Series Discretization for Knowledge Discovery", en Proc. of the Eleventh ACM SIGKDD international Conference on Knowledge Discovery in Data Mining (Chicago, Illinois, USA, August 21 - 24, 2005). KDD '05. ACM Press, New York, NY, 660-665, (2005).
10. Tan P., Steinbach M., Kumar V.: *Introduction to data mining*, Addison-Wesley, (2006).

# Constraint Programming, Planning, and Scheduling