Application of Time Series Discretization Using Evolutionary Programming for Classification of Precancerous Cervical Lesions

Héctor-Gabriel Acosta-Mesa[1], Fernando Rechy-Ramírez[1], Efrén Mezura-Montes[1], Nicandro Cruz-Ramírez[1] and Rodolfo Hernández Jiménez[2]

[1] School of Physics and Artificial Intelligence, Department of Artificial Intelligence, Universidad Veracruzana, Sebastián Camacho # 5, 91000, Xalapa, Ver. México.
heacosta@uv.mx, frechyr@hotmail.com, emezura@uv.mx , ncruz@uv.mx

[2] Obstetrician and Gynaecologist, Diego Leño # 22, C.P. 91000, Xalapa, Ver. México.
roheji@msn.com

Correspondence to:
Dr. Hector-Gabriel Acosta-Mesa
E-mail address: heacosta@uv.mx, heacosta@live.com.mx
Postal address: Department of Artificial Intelligence, Universidad Veracruzana
Sebastian Camacho 5, Col. Centro, C. P. 91000, Xalapa, Veracruz, Mexico.
Tel: +52 228 817-28-55

Abstract

In this work, we present a novel application of time series discretization using evolutionary programming for classification of precancerous cervical lesions. The approach optimizes the number of intervals in which the length and amplitude of the time series should be compressed, preserving the important information for classification purposes. Using evolutionary programming, the search for a good discretization scheme is guided by a cost function which considers three criteria: the entropy regarding the classification, the complexity measured as the number of different strings needed to represent the complete data set, and the compression rate assessed as the length of the discrete representation. This discretization approach is evaluated using a time series data based on temporal patterns observed during a classical test used in cervical cancer detection; the classification accuracy reached by our method is compared with the well-known times series discretization algorithm SAX and the dimensionality reduction method PCA. Statistical analysis of the classification accuracy shows that the discrete representation is as efficient as the complete raw representation for the present application, reducing the dimensionality of the time series length by 97%. This representation is also very competitive in terms of classification accuracy when compared with similar approaches.

Keywords: Times Series Discretization, Evolutionary Algorithms, Classification, Cervical Cancer Detection.

1. Introduction

Many real-world applications related with information processing generate temporal data [14]. The temporal databases generated require enormous data storage. It is therefore desirable to compress this information while maintaining the most informative features. Previous work on this topic has been mainly focused on data compression. However, they do not rely on significant information measured with entropy [13, 15]. In those approaches, the dimensionality reduction is given by the transformation of time series of length N into a dataset of n coefficients, where $n < N$ [10]. The two main characteristics of a time series discretization scheme are: the number of segments in which the time series length has to be partitioned (word size) and the number of intervals required to represent its amplitude expressed by continuous values (alphabet). Fig. 1 shows a time series with a grid that represents the cut points for word size = 9 and alphabet = 7. Using this transformation each time series is discretized and represented as a string.

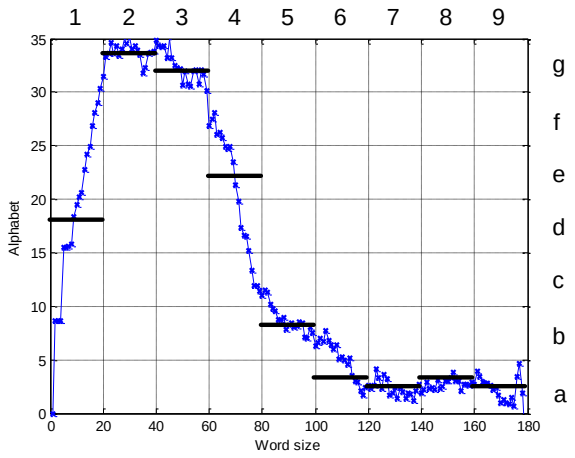


Fig. 1. Time series discrete representation. The time series length has been arbitrarily divided in nine segments of equal length. The continuous values of the time series amplitude are discretized in seven intervals of equal length as well. In this case the time series is represented as the string: d g g e b a a a a.

Among the approaches proposed to deal with time series data discretization we find those which work with one time series at a time, such as the one proposed by Mörchen [17]. His algorithm is centered on the search of persistent states (the most frequent values) in time series. However, such states are not common in many real-world applications for time series. Another representative approach was proposed by Dimitrova [7], where a multi-connected graph representation for time series was employed. The links between nodes have Euclidean distance values which are used under this representation to eliminate links in order to obtain a path that defines the discretization scheme. Nonetheless, this way to define the discretization process could be disadvantageous because not all the time series in a data set will necessarily have the same discretization scheme.

Keogh [15] proposed the Symbolic Aggregate Approximation (SAX) approach. This algorithm is based on the Piecewise Aggregate Approximation (PAA), a dimensionality reduction algorithm [11]. After PAA is applied, the values are then transformed into categorical values through a probability distribution function. Although SAX is an

improvement of PAA, both algorithms require the alphabet and the word size as inputs, which is their main disadvantage because it is not clear how to define them for a given time series dataset.

There are other approaches based on search algorithms, for example García-Lopez [6] proposed EBLA2, which in order to automatically find the word size and alphabet performed a greedy search looking for entropy minimization. The main disadvantage of this approach is the sensitivity of the greedy search leads it to get trapped in local optima. Therefore, they used simulated annealing as a search algorithm and the results were improved. Finally, in [1], a genetic algorithm was used to guide the search; however the solution was incomplete in the sense that the algorithm considered the minimization of the alphabet and the word size as two sequential and independent processes. In this way some solutions could not be generated and the obtained solution is not global. In order to avoid it, we developed an algorithm that automatically finds both parameters at the same time [5].

This is the most important contribution of our approach, since most of the discretization algorithms require, as an input, the parameters of word size and alphabet [12, 15]. However, in real-world applications it might be very difficult to know in advance their best values. Hence, their definitions require a careful analysis of the time series data set. In this work, we introduce the main ideas behind our discretization approach in which both the word size and the alphabet are calculated automatically, and how this approach can be applied to the medical field in the classification of precancerous cervical lesions.

Cervical cancer is the second leading cause of death for women worldwide. If it is detected early, the probability of cure is very high. After Pap smear test, colposcopy is the most used technique to diagnose this disease due to its higher sensitivity and specificity. Colposcopy allows us to visualize the uterine cervix using a microscope fitted with a light source. During the colposcopic test, the appearance of the cervix is observed while a solution of acetic acid is spread on the epithelium, which produces a change from the usual pink tissue to a whitish color due to the coagulation of proteins in the cellular nucleus. This phenomenon is called acetowhitening, its effect is more evident at the wavelet of 525 +/- 15 (green) due to the haemoglobin absorption made by the stroma. Acetowhitening disappears in less than 10 minutes and is more evident in precancerous lesions due to an altered nucleus to cytoplasm ratio (Fig. 2).

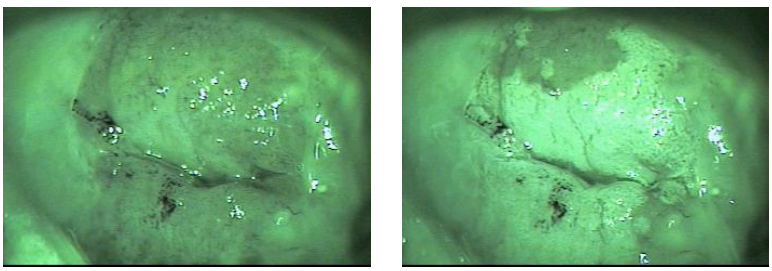


Fig. 2. Aceto-white effect. Cervical tissue viewed through an optical green filter before (left picture) and after (right picture) the application of acetic acid.

The primary problem with this technique is the intrinsic subjectivity of the test, i.e. the amount and speed of color change perceived could be different for various observers; this fact may produce high variability on the diagnoses made by experts. Therefore

changes in the mechanisms to quantify the amount of acetowhite change and the speed changes are needed to improve the test. Some researchers have suggested using the temporal patterns intrinsic to the color changes, which we called Aceto-White Response Functions (AWRF).

Costas J. Balas et al. proposed the use of spectroscopy to study the correlation between aceto-white patterns and precancerous cervical lesions [2]. Although some efforts have been made to characterize precancerous cervical lesions using aceto-white temporal patterns, to the best of our knowledge there is not a complete understanding of how to automatically analyze colposcopic images using aceto-white temporal patterns for classification of cervical tissue. In our previous work [3,4], we compared the shape of the temporal patterns to establish relationships among similar shapes, and the correlation of those patterns with certain types of tissue.

In the present work we apply to these temporal patterns our discretization approach in order to compress and analyze the structural properties of the intrinsic dynamics involved in the aceto-white phenomenon. In order to find a competitive discretization scheme that provides a suitable word size and alphabet, and considering its simplicity with respect to other evolutionary algorithms, evolutionary programming (EP) is adopted as a search algorithm: no recombination and parent selection mechanisms are performed and just mutation and replacement need to be designed. In a previous work the efficiency of this approach was assessed using 20 time series databases of the UCR Time Series Classification/Clustering repository [5].

The main contribution of this work is the application of our discretization approach to a medical domain in order to contribute to the solution of one of the most important health problems. The contents of this paper are organized as follows: Section 2 introduces the context in which the application is motivated. After that, Section 3 exposes the main ideas behind the proposed discretization approach. Section 4 presents results. Finally, Section 5 draws some conclusions and prospects for future work.

2. Materials

2.1. Data acquisition

Two hundred women were included in this study, within the total number of patients, in 100 cases a tissue sample or biopsy was obtained because some changes in the appearance of the cervical epithelium were observed by the colposcopist and these alterations led to suspicion of a lesion. In the other 100 patients, the specialist did not find changes that suggested the presence of a lesion and due to clinical protocols a biopsy was not taken. Of the total quota, 93 cases were positive for precursor lesions of cervical cancer and 107 negative. Before the test, the patients signed an informed consent.

Subsequently, during the colposcopic test a set of digital images were obtained. The acquisition was performed using a colposcope Vasconcellos CP-M1225 with an STC-N63BJ camera. Because in previous research [2] it has been reported that the acetowhitening effect is higher at the wavelet of 525 +/- 15 (green), the image acquisition was made using a green optical filter. The dimension of the images was 352 x 240 pixels with a sampling frequency of 1 frame / second. The images were stored as

separated files in the BMP format. A tool was developed for the acquisition and it was implemented in MATLAB 7.0. Before the application of three milliliters of 3% acetic acid solution, 10 images were obtained as a reference to calculate the percentage of color change of the tissue. Then, 180 images were acquired during a period of 3 minutes (60*3). When the acquisition process was completed, the colposcopist selected the region where the biopsy was obtained on one image of the sequence. As the images were acquired using a green filter, they did not have chromatic components other than different saturations of green hue. For simplicity, and given our previous experience in [3], image analysis was done in grayscale. The misalignment on the image sequences produced by subject´s movements were corrected using a normalized cross-correlation registration method [3,20].

2.2. Time series extraction

The colposcopic image sequence can be represented as a sequence of *t* 2D images $I_t(x,y)$ with acquisition time *t*. The color variation over time of each pixel in the image provides a time series. The resulting image sequence can be viewed as a 3D image block I(*x,y,t*) defined on the spatio-temporal domain. Let ImageBlock(*x,y,t*) represent a stack of t images of size (*x,y*). Thus there are (*x* * *y*) pixels, each of which is a time series of length t. Let *p(i,j)* represent the intensity of the pixel (i,j), i=1, ..., x, j=1, ..., y. The intensity value of each pixel over time is used to construct an AWRF (Fig. 3).

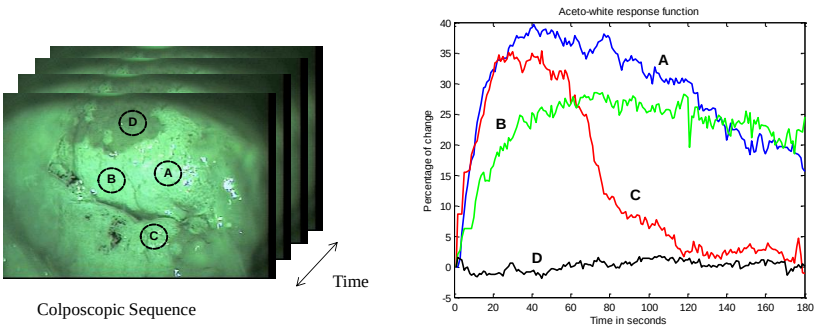


Fig. 3. Aceto-white response functions. At the right, temporal patterns observed on the pixels are denoted by the circles labeled on the left image. Different types of tissue produce AWRF with different percentage of change, speed of increase and decay.

As was mentioned on section 2.1, after the acquisition of the images, the colposcopist selected the region where a biopsy was obtained on one representative region of the tissue type. Using these regions of interest defined by the expert, the mean time series of each region was used to construct a database where each row contains the AWRF of each subject with their corresponding tissue type. Then, the size of our time series database (TSDB) is 200 rows (subjects) by 180 columns (time series points), plus one column that identifies the class label (+ o -). The classification task consists of finding similar temporal patterns with regular shapes belonging to the same type class.

2.3. PCA compression

Although the purpose of our approach is discretization and not only dimensionality reduction, a principal component analysis (PCA) was developed in order to compare the classification accuracy of both methods.

PCA was developed over the complete database of AWRF (TSDB). The objective of the analysis was to find the first *n* principal components (PC) which explain the shapes of the AWRF contained in the TSDB. The four principal components were selected to explain the 98.06% of the variance (see Fig. 4).

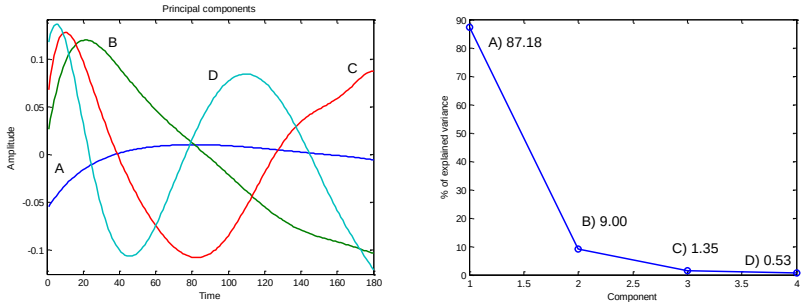
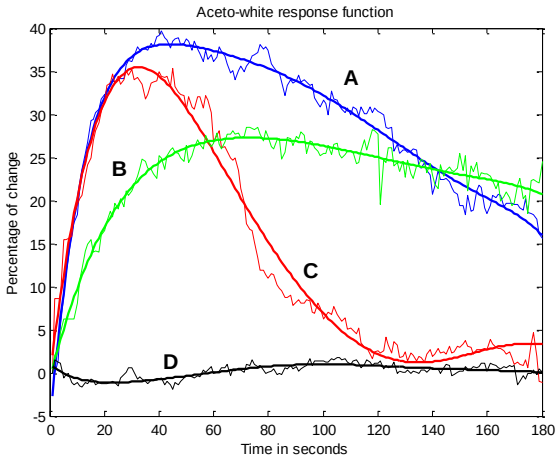


Fig. 4. Principal components. The four principal components obtained from the TSDB are shown at the left. Their corresponding percentage of variance explained by each component is shown at the right.

Consider the linear model: TSDB = θ*PC + e, where TSDB is the time series database of AWRF, PC is the matrix of regressors (represented by the four first principal components), θ is the parameter matrix corresponding to the projections of each AWRF over the PC, and e is a normally distributed noise vector. The parameter estimation can be easily done using linear regression. Eq. 1.

$$\theta = \frac{TSDB * PC}{PC * PC} \tag{1}$$

For illustrative purposes, Fig. 5. shows four representative AWRF and their corresponding adjusted models and parameters theta.



	θ_1	θ_2	θ_3	θ_4
A	670.0	96.5	18.6	7.1
B	408.2	4.2	0.1	-4.1
C	229.4	168.0	24.4	-48.2
D	-18.8	-7.7	-7.6	2.4

Fig. 5. AWRF fitted model. At the right the raw data of four AWRF is shown with their corresponding adjusted models. The values of the theta parameter for each AWRF over the four principal components is shown in the table at the right.

3. Methods

3.1. Time series discretization

The discretization process refers to the transformation of continuous values into discrete values. Formally, the domain is represented as $x \mid x \in \Re$ where $\Re$ is the set of real numbers. The discretization scheme is $D = \{[d_0, d_1]_1, [d_1, d_2]_2, \ldots, [d_{(n-1)}, d_n]_s\}$ where d_0 and d_n are the minimum and maximum values for x respectively, $d_i \mid i = 0 \ldots n$ are the limits of intervals (cut points) and s is called the discretization degree. Each tuple in D represents an interval (segment), where each continuous value x is mapped to one segment $[d_a, d_b]_j$ where $d_a \leq x < d_b$. For the case of time series, the discretization process has to be done in both characteristics, the time series length (word) and amplitude (alphabet). See Fig. 6.

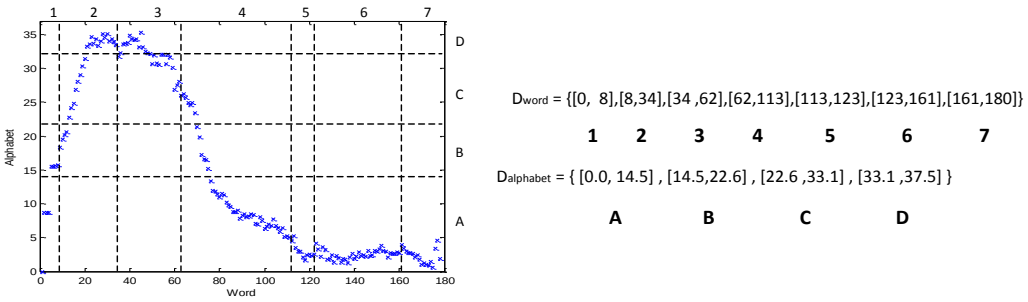


Fig. 6. Time series discretization scheme. For illustrative purposes the discretization scheme graphically shown at the left is described at the right using the representation explained above. Each tuple corresponds to one interval on word and alphabet.

Within our approach we use a modified version of the PAA representation [11]. PAA requires the word size and alphabet as an input value in order to generate the discretization scheme, which has the same length in all partitions (Fig. 1). In our algorithm each segment is calculated through the same idea as in PAA by using mean values, however partitions will not necessarily have equal lengths (Fig. 7). This difference can be stated as follows: let $TS = TS_1, \ldots, TS_n$ be a time series with length n , $\overline{TS} = \overline{TS_1}, \ldots, \overline{TS_s}$ the discretized version of TS with discretization scheme $D_{word} = \{[d_0, d_1]_1, \ldots, [d_{(n-1)}, d_n]_s\}$, where each $\{[d_k, d_{(k+1)}]_j\}$ is the time interval for the segment $\overline{TS_j}$, which is computed as the mean of the values on that partition. Eq. 2. See Fig. 7a.

$$\overline{TS_j} = \frac{1}{(d_{(k+1)} - d_k)} \sum_{i=d_k}^{d_{(k+1)}} TS_i \qquad (2)$$

Once $\overline{TS}$ is computed and according to the discretization scheme: $D_{alphabet} = \{[d_0, d_1]_1, \ldots, [d_{(n-1)}, d_n]_p\}$ each element $\overline{TS_j}$ is replaced by the discrete value (character) that corresponds to the segment $[d_a, d_b]_i$ where $d_a \leq \overline{TS_j} < d_b$. See Fig. 7b.

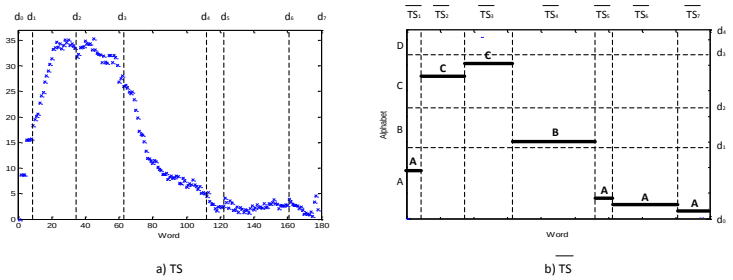


Fig. 7. Time series discretization. Given a discretization schema defined by D_{word} and $D_{alphabet}$ the continuous time series *TS* (a) is transformed to $\overline{TS}$ (b).

3.2. Evolutionary programming (EP)

EP is a simple but powerful evolutionary approach where evolution is simulated at species level, i.e., no crossover is considered [9]. Instead, asexual reproduction is implemented by a mutation operator. The main steps in EP are:

i) Population initialization.
ii) Evaluation of solutions.
iii) Offspring generation by mutation.
iv) Replacement.

From the steps mentioned above, the following elements must be defined so as to adapt EP to the time series discretization problem: a) solution encoding, b) fitness function to evaluate solutions, c) mutation operator and d) replacement mechanism.

3.2.1. Solution encoding

As in other evolutionary algorithms, in EP a complete solution of the problem must be encoded in each individual. This means that unlike other problem solving approaches such as state space search in which the solution is partially constructed step-by-step through search in a tree, in the evolutionary approach all the information needed for the solution is contained within a simple vector structure. In our case each individual represents a specific discretization scheme. A complete discretization scheme is encoded as shown in Fig. 8. The chromosome is divided into two sections; each one represents the word size (before the dotted line) and the alphabet. Word size is represented as an ordered list of integers, where consecutive cells under this vector represent the cut points (limits) of each interval. In a similar way the alphabet is represented as an ordered list of continuous values [19]. As mentioned above, the chromosome represents a complete discretization scheme. It can be seen as a grid constructed using each position of the vector as a cut point. For example, for the chromosome shown in Fig. 8 which corresponds to the discretization scheme shown in Fig. 6., the first segment of word size goes from position 0 to position 7, the second segment goes from position 8 to position 33, and so on. For the alphabet, the first discretized value goes from 0.0 to 14.5, the second from 14.5 to 22.6 and so on.

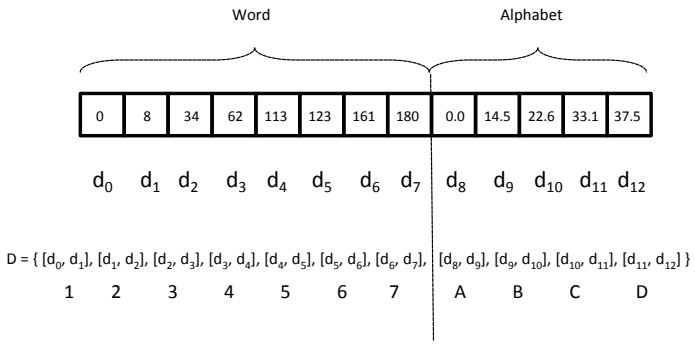


Fig. 8. Chromosome encoding. The chromosome contains information about the discretization scheme of the word size and the alphabet. For illustrative purposes, the values d_x shown inside de vector correspond to the discretization scheme shown in Fig. 5.

3.2.2. Fitness function

Different measures have been reported in the specialized literature to determine the quality of discretization schemes, such as information criterion [7], persistence state [17], information entropy maximization (IEM), information gain, entropy maximization, Petterson-Niblett and minimum description length (MDL) [13, 8]. Our fitness function, which aims to bias EP to promising regions of the search space, is based on three elements:

i) Classification accuracy (*accuracy*).
ii) Model complexity (*complexity*).
iii) Dimensionality reduction rate (*compression*).

These three values are normalized and added into one single value using the relationship described in Eq. 3 for each individual *i* in the population (Pop).

$$Pop_i = \alpha * accuracy + \beta * complexity + \gamma * compression \quad (3)$$

where: α, β *and* γ are the weights whose values determine the importance of each element.

The whole evaluation process for a given individual, i.e., computing the quality of a discretization scheme, requires the following steps: 1) The application of the discretization scheme over the complete time series database TSDB (*m* x *n*) to obtain $TSDB_d$ (*m* x *n*'), which is the discretized version of TSDB, where *m* is the number of rows (subjects on this application) and *n* is the time series length, then $n' < n$. 2) From $TSDB_d$, the computation of the number of different strings (*NDS*) generated by this specific discretization scheme. 3) The construction of a confusion matrix *CM* of dimensions (*NDS* x *C*), where C is the number of classes on the database. See Table 1.

		C	
	String	**-**	**+**
1	A A B C	43	0
2	A B B D	31	3
3	B B C D	13	10
4	B C B A	14	14
5	C C C B	6	30
6	D B C A	0	36
		107	**93**

Table 1. Confusion matrix (CM). For illustrative purposes a CM of a hypothetical discretization scheme applied to the TSDB of the present application is shown. For this case NDS = 6 and C=2. It is important to remember that our database has 200 observations (m), 107 for negative class and 93 for positive class.

Each cell of CM contains the number of times that one particular string was found in the database belonging to each class value. For an ideal discretization scheme each string has to be associated only to one class, otherwise the uncertainty on the classification stage will be increased.

The first element of Eq. 1 (*accuracy*) is computed through entropy calculation over the columns of the CM matrix. Entropy is an information measure, it tells us how reliable a variable is to predict the value of other variables. The lower the entropy, e.g. zero, the more informative the variable is. Entropy is computed as indicated in Eq. 4.

$$accuracy = \sum_{i=1}^{NDS} - \sum_{j=1}^{C} CM(P_{i,j}) * \log_2 CM(P_{i,j}) \quad (4)$$

Where $CM(P_{i,j})$ is the probability of the occurrence of the string *i* on the class *j*. Higher accuracy is achieved when strings belong only to one class due to the product of larger probability values. Strings with the same probabilities of belonging to both classes represent the worst case in terms of usefulness for prediction. For example, on table 1 string number 2 has lower entropy than string number 4. String number 1 has zero entropy, i.e., knowing the value of the string we can predict the class with total accuracy.

Using only entropy as fitness criteria, the generation of one different string per each observation in the raw database can be encouraged, in order to avoid this, the second term of the fitness function penalizes this aspect, assessing the complexity of the discretization scheme on the database as the number of different strings generated. In an ideal situation we would expect to have one string per class, in which case the complexity is zero. See Eq. 5. As a way of standardization and in order to keep the values on this term on the same range of those of the entropy term, the NDS minus the number of classes is divided by the number of observations on the database plus the number of classes.

$$complexity = (NDS - C) / (m + C) \quad (5)$$

Finally, the third element of the fitness function takes into account the dimensionality reduction of the solution, assessed as the number of partitions (word size) that a

particular discretization scheme has. For the sake of standardization this value is divided by two times the original length the time series. See Eq. 6.

$$compression = (n'/(2 * n)) \quad (6)$$

In summary, the first element represents how well a particular individual (discretization scheme) is able to correctly classify the database, the second element evaluates the complexity of the representation in terms of different patterns needed to encode the data, and the third element is a measure of the compression rate reached using a particular discretization scheme.

3.2.3. Mutation operator

The mutation operator is applied to every individual in the population in order to generate one offspring per individual. The number of changes to be made on an individual is controlled by the parameter $N\_MUT \in [1, 2, 3]$. Each change consists on randomly choosing two positions (one for word and one for alphabet) of the vector shown in Fig. 8. The selected positions are filled with new valid values and inserted in the corresponding place to maintain the vector in ascending order. Fig. 9.

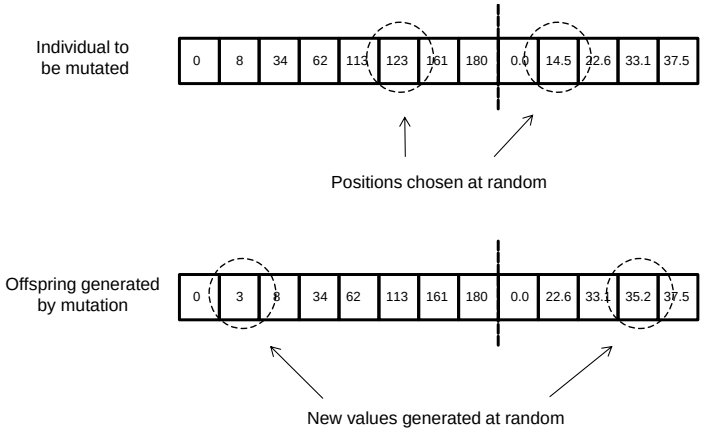


Fig 9. Mutation operator. Mutation is applied by randomly choosing one position (cut point) on the word representation (before dotted line) and one position of the alphabet (after dotted line).The new value is generated randomly and inserted in the new position according to the ascending order.

3.2.4. Replacement mechanism

The replacement mechanism consists in sorting the current population and their offspring by their fitness value., letting the first half survive for the next generation while the second half is eliminated.

The pseudocode of the EP algorithm is shown in *EP Algorithm*. The initial population (*Pop*) of *popsize* individuals is generated at random. Then, each individual generates one offspring by the mutation procedure described above. The set of current individuals *Pop* and the set of *Offspring* are merged into one new set called *Pop'*. This new set is sorted based on fitness. The first-half of individuals remains for the next generation. The process finishes when a given number of generations (*maxgen*) is reached [5]. The best discretization scheme is then used to discretize the database to be classified using

the K-nearest neighbors (k-nn) algorithm [10] in order to assess performance in terms of error rates.

EP Algorithm % Pseudocode

```
1: Define popsize, maxgen,          % Input: Population size and Number of generations
2: Pop = ∅                          % Initial population set is empty
3: for i =1 to popsize do           % Creating the initial population
4:       Pop(i) = Valid Scheme()    % A valid individual is generated at random
5: end for
6: for i = 1 to maxgen do
7:       Offspring = ∅              % Offspring set is empty
8:       for j = 1 to popsize do
9:          Offspring(j) = Mutation(Pop(j))      % Create a new valid individual
10:      end for                                 % by mutation
11:      Pop' = Replacement(Pop + Offspring)     % Select the best ones
12:      Pop=Pop'                                % Population update
13: end for
```

4. Results

The discretization scheme was obtained using the EP algorithm explained above using the following parameters tuned using the IRACE tool [16], which implements the statistical Iterated Race method to find suitable parameter values for evolutionary algorithms: *popsize* = 272 and *maxgen* = 50, $\alpha = 0.706$, $\beta = 0.1403$ and $\gamma = 0.007$. As can be seen, in our experiments accuracy had priority over complexity and compression. Fig. 10 shows four different discretization schemes obtained through the search. Discretization schemes with high partitions on word and alphabet reach better accuracies, however their fitness functions increases due to the increase on the complexity and compression terms.

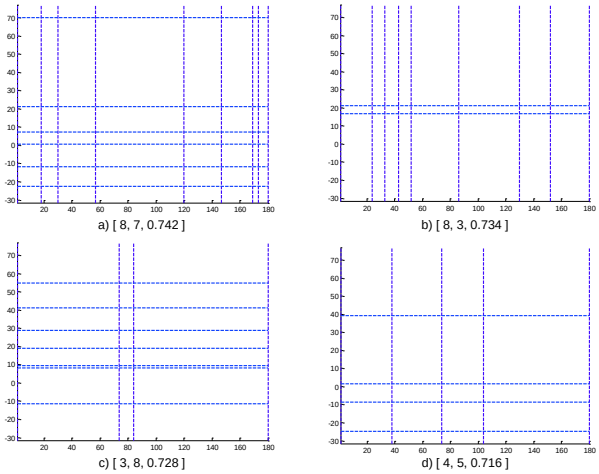


Fig. 10. Discretization schemes. Four solutions obtained through the evolution of the algorithm are shown. The word size, alphabet size and fitness function of each discretization scheme are shown in square brackets.

The quality of the solutions found by the EP algorithm was computed by using the best discretization scheme obtained for a set of ten independent runs of the k-nn. The distance measure used in the k-nn algorithm was Euclidean distance. The classification accuracy of the algorithm over the colposcopic time series database was assessed using stratified holdout (2/3 of the data for training and 1/3 for testing).

Ten runs were done using different samples of data for each run (using the same samples), the algorithm was executed five times in order to avoid local minima due to random initialization. The best solution of these five executions is reported per each of the ten runs. Each discrete representation was evaluated in terms of classification error using K=[1,3,5,7]. Table 2. The only statistical difference in error rates found using the non-parametric ANOVA test (Kruskal-Wallis) at 95% confidence, was between the errors reached by K=1 and K=7.

	Discretization Scheme		Error			
Run	**Word Size**	**Alphabet**	**K=1**	**K=3**	**K=5**	**K=7**
1	3	8	0.348	0.288	0.242	0.197
2	7	7	0.379	0.394	0.394	0.364
3	3	11	0.455	0.364	0.515	0.394
4	6	7	0.364	0.288	0.379	0.348
5	3	7	0.439	0.439	0.379	0.439
6	8	3	0.394	0.318	0.303	0.333
7	8	3	0.288	0.333	0.303	0.242
8	3	9	0.409	0.288	0.303	0.258
9	5	5	0.303	0.348	0.424	0.303
10	3	9	0.318	0.333	0.333	0.348
Mean	**5**	**7**	**0.370**	**0.339**	**0.358**	**0.323**
Std			0.056	0.050	0.078	0.074

Table 2. Classification error. The database was segmented in ten partitions using stratified holdout. EP discretization algorithm was run in each dataset. The word size and alphabet found by the algorithm is shown in each case. The error rates (scale 0 to 1) reached by each discretization scheme was evaluated for K=[1,3,5,7].

The performance of the EP algorithm was compared with one of the most popular time series discretization algorithms (SAX), and the dimensionality reduction method (PCA). The classification accuracy reached was computed using the same classification methodology and data sets explained above. The classification error was computed using k=7 as it is the parameter which reaches the lower mean error in our experiments. For the case of SAX method and because this algorithm requires the word size and the alphabet as parameter input, the discretization was made using the word size and alphabet found by our EP algorithm. PCA compression was made using the methodology explained in section 2.3. Classification accuracy reached using raw data is also shown as a reference. See Table 3. Performance of Raw, EP and PCA are statistically equivalent.

	Error			
Run	**Raw**	**EP**	**SAX(EP)**	**PCA**
1	0.258	0.197	0.424	0.287
2	0.333	0.364	0.500	0.333
3	0.333	0.394	0.394	0.393
4	0.348	0.348	0.500	0.363
5	0.364	0.439	0.455	0.424
6	0.242	0.333	0.394	0.333
7	0.242	0.242	0.424	0.287
8	0.318	0.258	0.333	0.333
9	0.333	0.303	0.364	0.348
10	0.348	0.348	0.424	0.363
Mean	**0.312**	**0.323**	**0.421**	**0.346**
Std	0.046	0.074	0.054	0.043

Table 3. Performance results. Error rates reached by the algorithm 7-nn using raw data, EP discretization, SAX(EP) discretization and PCA compression. Last row shows the mean and standard deviation.

5. Discussion

5.1. Performance evaluation

The classification accuracy reached using the EP discretized data was statistically equivalent to that obtained using the continuous data, with the advantage that the space needed for its representation is smaller than the continuous one, a reduction of almost 97% in the word size average (5/180). Although PCA representation is also statistically equivalent, it is important to remember that this is a dimensionality reduction method and not a discretization method. The PCA representation contains continuous values that do not have direct interpretation, and also restricts its use to other classification approaches that requires categorical data. In the case of SAX representation, it has poor performance because under this approach all the partitions on word and alphabet has the same length, and the parameters found by EP were not necessarily suitable for this approach.

Other particularity of the EP approach is the fact that the solution found by the algorithm can be biased by the criteria of Accuracy (α), Complexity (β), and Compression (γ), letting us a solution according to our requirements, or can be automatically found using a tool as IRACE. It is important to remark that in our algorithm the partitions created for word and alphabet have different sizes; it makes the approach much more flexible to find better discretization schemas. The comparison of the performance of the EP algorithm against other similar approach that uses variable partitions [1,6] is not included in the table of results because it was done in [5] using 20 different databases. The EP search strategy showed to be more efficient than GENEBLA algorithm.

The classification errors reached by the k-nn algorithm using different values for K, suggest that the classes are not well separated as the algorithm needs high K values to decrease the error. A possible explanation for this phenomenon could be that aceto-

white epithelium changes gradually from healthy (-) to unhealthy (+) and under this transition there is not a clear separation among the AWRF's shape. A more detailed specification of classes that make subdivision under positive and negative classes could clarify which subclasses can be more suitable for classification. On the other hand, as the k-nn considers the discrete AWRF (string) as a complete pattern, it is no way to differentiate if particular segments (characters) of the string are more informative to improve the classification. On this sense, the next section introduces some keys to make this analysis.

5.2. AWRF feature extraction

Although the main objective of our experiments was to investigate how the discretization process allows us to reduce the dimensionality of the database in terms of word size and alphabet while the classification accuracy is maintained on this application, we also analyzed how the discrete representation lets us identify the most important segments of the time series that helps to properly classify an observation. In order to increase the compression rate and also to identify the most informative parts of the time series length, each segment of the word was evaluated in terms of its information content. The idea behind this analysis was to assess the homogeneity (in terms of class values) of each bin created along the alphabet. For example, let us consider the discretization scheme shown in Fig. 11.

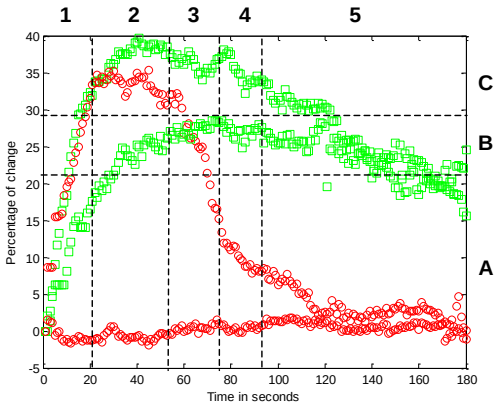


Fig 11. Information content in time series segments. A discretization scheme of 5 word segments [1,2,3,4,5] and 3 alphabet segments [A,B,C] is shown over four time series belonging to two classes (o and □).

For illustrative purposes we have included four time series, two belonging to class **o** and two belonging to class □. Considering the discretization scheme shown in the figure we can see that different cells are created by the intersections between word and alphabet cut points. Some of the cells contain observations of one class and some other contain observations of both classes. For classification purposes the cells that contain only one class are more informative, since knowing them we can infer the class, that is to say, those cells has low entropy. Under this approach it can be said that the segment of the discretization scheme which contains cells with low entropy are more informative for classification purposes.

For example, in segment 1 the informative content is poor because on its three cells the classes are mixed. Segment 2 is more informative than segment 1 since only in one of

three cells the classes are mixed, however segment 4 has zero entropy because each cell only contains observations of one class. The segment information can be assessed using Eq. 7.

$$Segment\_Info_k = \sum_{i=1}^{alphabet} -\sum_{j=1}^{C} P_{i,j} * \log_2(P_{i,j}) \quad (7)$$

Where C is the number of classes, and $P_{i,j}$ is the probability of class j to appear on the cell formed by the alphabet value i on the segment k. Once the segment information of each partition is computed, a sorted list of segment entropies can be constructed to rank its informative significance. Fig. 12. In order to assess the contribution of each segment in the classification performance, the classification algorithm was run using the k most informative segment from the sorted list. The error rate reached considering the first k segments with the highest entropy was plotted on the same figure. As it can be appreciated, non all the segments provide additional information to reduce classification error, then those segments could be eliminated from the representation. On the other hand, some segments with high entropy contribute to improve accuracy. This fact has to be further investigated and is considered as part of the future work explained in the next section.

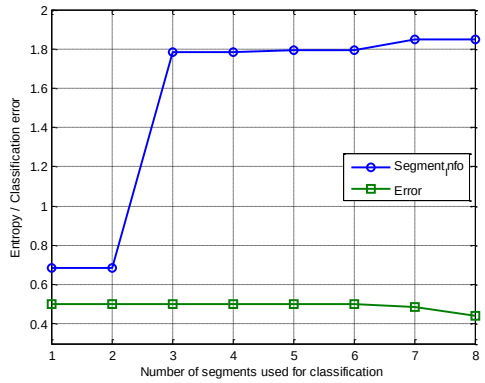


Fig. 12. Segment information rank versus classification error rate. Consider a discretization scheme proposed by the algorithm with word size = 8. The graph shows the ordered segment information of each segment and how the error rate behaves when some segments are discarded from the classification process. The inclusion of segments was accumulative in the sense that the error plotted in 1 represents the inclusion of the first most informative segment; the next point refers to the inclusion of two first segments and so on.

5.3. Clinical application

As we said before, cervical cancer is the second leading cause of death for women worldwide. If it is detected early, the probability of cure is very high. The present work is one step towards the aim to construct an expert system capable to automatically segment a colopocopic image in order to guide the medical expert in the selection of the best region from which to take a biopsy [2,3]. The general structure of the system under construction considers four main procedures to obtain a segmented image: Image registration, Time series representation, Knowledge acquisition and Machine learning. Fig. 13.

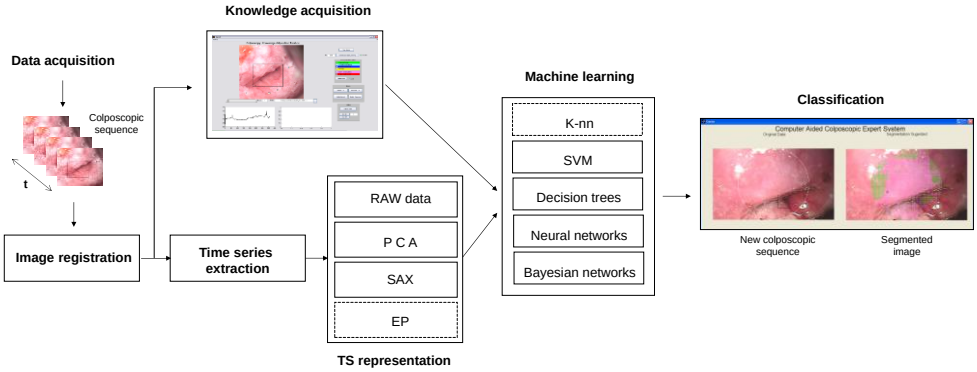


Fig. 13. Computer aided colposcopic expert system structure. The main modules of the system are represented by boxes. The procedures developed by each module can be performed using different approaches. In dotted lines are shown the sub-modules developed as result of the present proposal.

After colposcopic secuence acquisition, it is necessary to correct the motion and deformations occurred during the acquisition time. This realignment is important to guarantee that the temporal patterns belong to the same region [21]. These temporal patterns are used to construct a database using the raw data and they are represented using different approaches in order to extract the maximum information; in the present work we propose the EP representation.

In order to get examples from which to learn from the expertise of the colposcopist, a knowledge acquisition module was developed and used as explained in section 2.1. Finally, using machine learning algorithms, the system is capable to learn from the examples, constructing a model able to discriminate new cases based on the characteristics of the temporal patterns. This information could be used to segment the complete image classifying the type of tissue according to its AWRF [4].

6. Conclusions and future work

We presented a novel application of time series discretization algorithm based on EP. The proposed algorithm was able to automatically find the parameters for a suitable discretization scheme considering the optimization of accuracy, complexity and compression rate. A simple mutation operator was able to sample the search space by generating new and competitive solutions. Our EP algorithm is easy to implement and our results show that the EP algorithm is able to create suitable discretization schemes that reaches statistically equivalent classification error than the raw data.

On the particular application of classification of precancerous cervical lesions using a temporal approach, we showed that the discretized version of the continuous raw data is able not only to reduce the dimensionality of data but also to maintain the classification accuracy. This research is one step toward the development of an automatic classification system for cervical cancer detection. As a difference with other well-known time series discretization algorithms like SAX, EP automatically finds a suitable parameter for word and alphabet size. Although PCA representation was as competitive as EP, in terms of classification accuracy and compression, it doesn't produce a discrete representation and the interpretation of this representation is not straightforward.

As a future work we consider a further analysis of the EP algorithm such as the effect of the weights in the search as well as the number of changes in the mutation operator. Other nature-inspired approaches like Particle-Swarm Optimization need to be tested. Pareto dominance will be explored with the aim of dealing with the three objectives considered in the fitness function [18]. Also, the discretized representation of the temporal patterns could let the application of other classification techniques (besides k-nearest neighbors) that work with categorical data, e.g. decision trees or Bayesian networks.

Finally, the segment information extraction can be extended to identify not only the most informative segments, but also the most informative cells. Using this information a graphical model of classification such as decision trees can be constructed; where the internal nodes are the identification of the word segment, the arcs are the values of the alphabet segment that identifies a particular cell, and the leaf nodes are the class values. Under this approach we could easily show graphically the most important segments on the time series. Fig. 14.

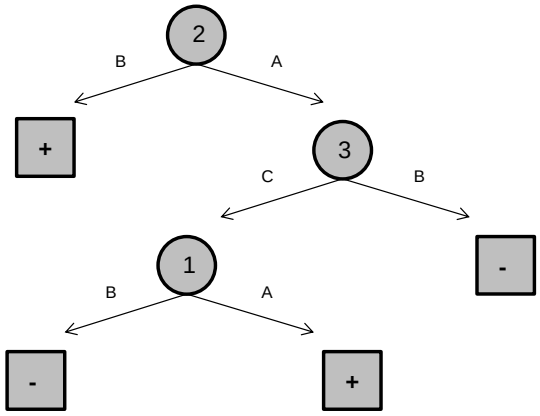


Fig. 14. Time series classification using Decision Trees (DT). Given a discretization scheme and using segment information assessed using entropy, the time series classification can be done using DT, representing word elements as internal nodes and alphabet values as arcs.

Acknowledgments

The first author thanks the Mexican National Council for Science and Technology for the economic support of this project under the research grant: Fondo Sectorial de Investigación en Salud y Seguridad Social SSA/IMSS/ISSSTE-CONACYT (SALUD-2008-01-86566). The authors also thank the valuable collaboration of M.Sc. Karina Gutierrez Fragoso.

References

[1] García-López Daniel-Alejandro, and Acosta-Mesa Héctor-Gabriel. Discretization of Time Series Data set with a Genetic Search. Lecture Notes in Artificial Intelligence. Springer 2009 Vol. 5845. pp. 201-212.
[2] C. Balas, A novel optical imaging method for the early detection, quantitative grading, and mapping of cancerous and precancerous lesions of cervix. IEEE Trans. Biomed. Eng. 2001, vol. 48, no. I, pp. 96-104.

[3] Acosta-Mesa H., Cruz-Ramírez N., and Hernández-Jimenez R., Aceto-white temporal pattern classification using k-NN to identify precancerous cervical lesion in colposcopic images, Comput. Bioi. Med. 2009. Vol. 39. pp. 778-784.
[4] Acosta-Mesa Héctor-Gabriel, Cruz-Ramírez Nicandro, Gutierrez-Fragoso Karina, Barrientos-Martínez Rocio- Erandi and Hernández-Jimenez Rodolfo. Assessing the Possibility of Identifying Precancerous Cervical Lesions using Aceto-White Temporal Patterns, Decision Support Systems Advances in, Ger Devlin (Ed.), INTECH 2010. ISBN: 978-953-307-069-8.
[5] Rechy-Ramírez Fernando, Acosta Mesa Héctor-Gabriel, Mezura-Montes Efrén, and Cruz-Ramírez Nicandro. Time Series Discretization Using Evolutionary Programming. Lecture Notes in Computer Science, 2011. Volume 7095, Advances in Soft Computing, pp. 225-234.
[6] Acosta-Mesa H.G., Cruz-Ramírez Nicandro, García-Lopez Daniel-Alejandro: Entropy Based Linear Approximation Algorithm for Time Series Discretization. In: Advances in Artificial Intelligence and Applications. Research in Computers Science. 2008 Vol. 32, pp. 214-224.
[7] Dimitrova E.S., McGee J., Laubenbacher E.: Discretization of Time Series Data. eprint arXiv 2005: q- bio/0505028.
[8] Fayyad U., Irani K.: Multi-Interval Discretization of Continuous-Valued Attributes for Classification Learning. In: Proceedings of the 13th International Joint Conference on Artificial Intelligence 1993. Machine Learning pp. 1022-1027.
[9] Fogel L.: Intelligence Through Simulated Evolution. Forty years of Evolutionary Programming (Wiley Series on Intelligent Systems) 1999. ISBN-13: 978-0471332503.
[10] Han J., and Kamber M.: Data Mining: Concepts and Techniques (The Morgan Kaufmann Series in Data Management Systems) 2001. ISBN-13: 978-1558604896.
[11] Keogh E., Chakrabarti K., Pazzani M., S. Mehrotra: Locally Adaptive Dimensionality Reduction for Indexing Large Time Series Databases, ACM Trans. Database Syst 2002. Journal ACM Transactions on Database Systems (TODS). Vol. 27 Issue 2, pp 188-228.
[12] Keogh E., S. Lonardi, Ratanamabatana C.A.: Towards parameter-free data mining. In KDD '04 proceedings of Tenth ACM SIGKDD international Conference on Knowledge Discovery and Data Mining 2001. Pp 206-215.
[13] Kurgan L., and Cios K.: CAIM Discretization Algorithm, IEEE Transactions On Knowledge And Data Engineering. Vol. 16, no. 2, february 2004, pp. 145-153.
[14] Last M., Kandel A., and Bunke H.: Data mining in time series databases, World Scientific Pub Co Inc, Singapore 2004. Vol. 14, pp. 1-22.
[15] Lin J., Keogh E., Lonardi S., y Chin B.: A symbolic representation of time series, with implications for streaming Algorithms, In DMKD '03 Proceedings of the 8th ACM SIGMOD Workshop on Research Issues in Data Mining and Knowledge Discovery 2003. pp 2-11.
[16] López-Ibáñez, M., Dubois-Lacoste, J., Stützle, T., and Birattari. M., The irace package, Iterated Race for Automatic Algorithm Configuration. Technical Report TR/IRIDIA/2011-004, IRIDIA, Université Libre de Bruxelles, Belgium, 2011.
[17] Mörchen F., y Ultsch A.: Optimizing Time Series Discretization for Knowledge Discovery. In: Proceeding of the Eleventh ACM SIGKDD international Conference on Knowledge Discovery in Data Mining 2005. Pp.660-665.
[18] Meyarivan T., Deb Kalyanmoy, Pratap A., Agarwal S., Pratap A., and Agarwal S. : A Fast and Elitist Multiobjective Genetic Algorithm: NSGA-II. IEEE Transactions on evolutionary computation 2002. Vol. 6, Issue 2, pp. 182-197.

[19] Nanh S.C., and Chaochang Chiu: An adapted covering algorithm approach for modeling airplanes landing gravities. Department of Information Management, Yuan Ze University, 135 Far East Rd., Chung-Li 320, Taiwan, ROC. Expert Systems with Applications 26 (2004) 443450.
[20] Zitova B and Flusser J 2003 Image registration methods: a survey Image and vision computing 21 11 977-1000.
[21] Efrén Mezura-Montes, Héctor Gabriel Acosta-Mesa, Darío-del-Sinaí Ramírez-Garcés, Nicandro Cruz-Ramírez and Rodolfo Hernández-Jiménez, An image registration method for colposcopic images, Computational and Mathematical Methods in Medicine, Vol. 2013, Article ID: 285962, DOI: 10.1155/2013/285962, September 2013.