Pattern Recognition Letters 112 (2018) 49–55

Contents lists available at ScienceDirect

Pattern Recognition Letters

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

Extended-alphabet finite-context models

João M. Carvalho[a,*], Susana Brás[a,b], Diogo Pratas[a], Jacqueline Ferreira[c,d], Sandra C. Soares[c,e,f], Armando J. Pinho[a,b]

[a] Institute of Electronics and Informatics Engineering of Aveiro, University of Aveiro, Portugal
[b] Department of Electronics Telecommunications and Informatics of Aveiro, University of Aveiro, Portugal
[c] Department of Education and Psychology, University of Aveiro, Portugal
[d] IBILI, Faculty of Medicine, University of Coimbra, Portugal
[e] CINTESIS-UA, University of Aveiro, Portugal
[f] Department of Clinical Neurosciences, Karolinska Institute, Stockholm, Sweden

ARTICLE INFO

Article history:
Received 29 June 2017
Available online 1 June 2018

ABSTRACT

The normalized relative compression (NRC) is a recent dissimilarity measure, related to the Kolmogorov complexity. It has been successfully used in different applications, like DNA sequences, images or even ECG (electrocardiographic) signal. It uses a compressor that compresses a target string using exclusively the information contained in a reference string. One possible approach is to use finite-context models (FCMs) to represent the strings. A finite-context model calculates the probability distribution of the next symbol, given the previous k symbols. In this paper, we introduce a generalization of the FCMs, called extended-alphabet finite-context models (xaFCM), that calculates the probability of occurrence of the next d symbols, given the previous k symbols. We perform experiments on two different sample applications using the xaFCMs and the NRC measure: ECG biometric identification, using a publicly available database; estimation of the similarity between DNA sequences of two different, but related, species – chromosome by chromosome. In both applications, we compare the results against those obtained by the FCMs. The results show that the xaFCMs use less memory and computational time to achieve the same or, in some cases, even more accurate results.

1. Introduction

Data compression models have been used to address several data mining and machine learning problems, usually by means of a formalization in terms of the information content of a string or of the information distance between strings [1–5]. This approach relies on solid foundations of the concept of algorithmic entropy and, because of its non-computability, approximations provided by data compression algorithms [6].

A finite-context model (FCM) calculates the probability distribution of the next symbol, given the previous k symbols. In this work, we propose an extension of the FCMs, which we call extended-alphabet finite-context models (xaFCM). Usually, these models provide better compression ratios, leading to better results for some applications, especially when using small alphabet sizes – and also by performing much less computations. We show this in practice for the ECG biometric identification and DNA sequence similarity. The source code for the compressor was implemented using Python 3.5 and is publicly available under the GPL v3 license.[1]

1.1. Compression-based measures

Compression-based distances are tightly related to the Kolmogorov notion of complexity, also known as algorithmic entropy. Let x denote a binary string of finite length. Its **Kolmogorov complexity**, $K(x)$, is the length of the shortest binary program x^* that computes x in a universal Turing machine and halts. Therefore, $K(x) = |x^*|$, the length of x^*, represents the minimum number of bits from which x can be computationally retrieved [7].

The **Information Distance** (ID) and its normalized version, the **Normalized Information Distance** (NID), were proposed by Bennett *et al.* almost two decades ago [8] and are defined in terms of the Kolmogorov complexity of the strings involved, as well as the complexity of one when the other is provided.

However, since the Kolmogorov complexity of a string is not computable, an approximation (upper bound) for it can be used by

* Corresponding author.
E-mail address: joao.carvalho@ua.pt (J.M. Carvalho).

[1] https://github.com/joaomrcarvalho/xafcm.

https://doi.org/10.1016/j.patrec.2018.05.026
0167-8655/

means of a compressor. Let $C(x)$ be the number of bits used by a compressor to represent the string x. We will use a measure based on the notion of *relative compression* [4], denoted by $C(x||y)$, which represents the compression of x relatively to y. This measure obeys the following rules:

- $C(x||y) \approx 0$ iff string x can be built efficiently from y;
- $C(x||y) \approx |x|$ iff $K(x|y) \approx K(x)$.

Based on these rules, the **Normalized Relative Compression** (NRC) of the binary string x given the binary string y, is defined as

$$\mathrm{NRC}(x||y) = \frac{C(x||y)}{|x|}, \tag{1}$$

where $|x|$ denotes the length of x.

A more general formula for the NRC of string x, given string y, where the strings x and y are sequences from an alphabet $\mathcal{A} = \{s_1, s_2, \ldots s_{|\mathcal{A}|}\}$, is given by

$$\mathrm{NRC}(x||y) = \frac{C(x||y)}{|x| \log_2 |\mathcal{A}|}. \tag{2}$$

2. Extended-alphabet finite-context models

2.1. Compressing using extended-alphabet finite-context models

Let $\mathcal{A} = \{s_1, s_2, \ldots s_{|\mathcal{A}|}\}$ be the alphabet that describes the objects of interest. An extended-alphabet finite-context model (xaFCM) complies to the Markov property, i.e., it estimates the probability of the next sequence of $d > 0$ symbols of the information source (depth-d) using the $k > 0$ immediate past symbols (order-k context). Therefore, assuming that the k past outcomes are given by $x_{n-k+1}^{n} = x_{n-k+1} \ldots x_n$, the probability estimates, $P(x_{n+1}^{n+d} | x_{n-k+1}^{n})$ are calculated using sequence counts that are accumulated, while the information source is processed,

$$P(w|x_{n-k+1}^{n}) = \frac{\nu(w|x_{n-k+1}^{n}) + \alpha}{\nu(x_{n-k+1}^{n}) + \alpha |\mathcal{A}|^d} \tag{3}$$

where $\mathcal{A}^d = \{w_1, w_2, \ldots w_{|\mathcal{A}|}, \ldots w_{|\mathcal{A}|^d}\}$ is an extension of alphabet $\mathcal{A}$ to d dimensions, $\nu(w|x_{n-k+1}^{n})$ represents the number of times that, in the past, sequence $w \in \mathcal{A}^d$ was found having x_{n-k+1}^{n} as the conditioning context and where

$$\nu(x_{n-k+1}^{n}) = \sum_{a \in \mathcal{A}^d} \nu(a|x_{n-k+1}^{n}) \tag{4}$$

denotes the total number of events that has occurred within context x_{n-k+1}^{n}.

In order to avoid problems with "shifting" of the data, the sequence counts are performed symbol by symbol, when learning a model from a string.

Parameter α allows controlling the transition from an estimator initially assuming a uniform distribution to a one progressively closer to the relative frequency estimator.

The theoretical information content average provided by the ith sequence of d symbols from the original sequence x, is given by

$$-\log_2 P(X_i = t_i | x_{id-k}^{id-1}) \text{ bits}, \tag{5}$$

where $t_i = x_{id}, x_{id+1} \ldots x_{(i+1)d-1}$.

As testbed applications, we perform ECG biometric identification and compute a similarity measure between DNA sequences;After processing the first n symbols of x, the total number of bits generated by an order-k with depth-d xaFCM is equal to

$$-\sum_{i=1}^{n/d} \log_2 P(t_i | x_{di-k}^{di-1}), \tag{6}$$

Table 1
FCM representation of the sequence *AAABCC*.

Context c	$\nu(A\|c)$	$\nu(B\|c)$	$\nu(C\|c)$	$\nu(c) = \sum_{a \in \mathcal{A}} \nu(a\|c)$
BC	0	0	1	1
CA	1	0	0	1
AB	0	0	1	1
CC	1	0	0	1
AA	1	1	0	2

Table 2
Proposed xaFCM representation of the sequence *AAABCC* (with $d = 1$). Notice that this model has exactly the same information as the one in Table 1.

Context c			
BC	C: 1	Total: 1	
CA	A: 1	Total: 1	
AB	C: 1	Total: 1	
CC	A: 1	Total: 1	
AA	A: 1	B: 1	Total: 2

where, for simplicity, we assume that $n(modd) = 0$.

If we consider a xaFCM with depth $d = 1$, then it becomes a regular FCM with the same order k. In that sense, we can consider that a FCM is a particular case of a xaFCM.

An intuitive way of understanding how a xaFCM works is to think of it as a FCM which, for each context of length k, instead of counting the number of occurrences of symbols of $\mathcal{A}$, counts the occurrences of sequences $w \in \mathcal{A}^d$. In other words, for each sequence of length k found, it counts the number of times each sequence of d symbols appeared right after it.

Even though, when implemented, this might use more memory to represent the model, an advantage is that it is possible to compress a new sequence of length m, relatively to some previously constructed model, making only m/d accesses to the model. This significantly reduces the time of computation, as we will show in the experimental results presented in Sections 3 and 4.

Since, for compressing the first k symbols of a sequence, we do not have enough symbols to represent a context of length k, we always assume that the sequence is "circular". For long sequences, specially using small contexts/depths, this should not make much difference in terms of compression, but as the contexts/depths increase, this might not be always the case.

Since the purpose for which we use these models is to provide an approximation for the number of bits that would be produced by a compressor based on them, whenever we use the word "compression", in fact we are not performing the compression itself. For that, we would need to use an encoder, which would take more time to compute. It would also be needed to add some side information for the compressor to deal with the circular sequences – but that goes out of scope for our goal.

2.1.1. Example

Let x be the circular sequence *AAABCC*. Using a regular FCM with $k = 2$ and $\alpha = 0.01$, we would build the model from Table 1 to represent x.

It is easy to notice that this representation can be implemented using an hash-table of strings to arrays of integers with fixed size (alphabet size +1). However, we propose a different alternative, which consists of building a hash-table of hash-tables. The reason for doing so is that often the number of counts of symbols for each context is very sparse, which would be a waste of memory. To represent exactly the same model, we would build the structure presented in Table 2.

Table 3
Proposed xaFCM representation of the sequence *AAABCC* (with $d = 2$).

Context c			
BC	CA: 1	Total: 1	
CA	AA: 1	Total: 1	
AB	CC: 1	Total: 1	
CC	AA: 1	Total: 1	
AA	AB: 1	BC: 1	Total: 2

For compressing the sequence x, relatively to itself, we would need $C(x||x)$ bits, where

$$C(x||x) = C(A|CC) + C(A|CA) + C(A|AA) + C(B|AA) + C(C|AB) + C(C|BC) \quad (7)$$

and,

$$C(A|CC) = C(A|CA) = C(C|AB) = C(C|BC) = -\log_2 \frac{1+0.01}{1+3\times 0.01} = 0.0283 \quad (8)$$

and

$$C(A|AA) = C(B|AA) = -\log_2 \frac{2+0.01}{1+3\times 0.01} = 1.007, \quad (9)$$

which means $C(x||x) = 2.1272$ or, in other words, it is possible to compress x relatively to itself using just 2.1272 bits.

Using a xaFCM, also with $k = 2$ and $\alpha = 0.01$, but with $d = 2$, we would build the model presented in Table 3 to represent x.

Therefore,

$$C(x||x) = C(AA|CC) + C(AB|AA) + C(CC|AB) \quad (10)$$

where,

$$C(AA|CC) = C(CC|AB) = -\log_2 \frac{1+0.01}{1+3^2\times 0.01} = 0.110 \quad (11)$$

and

$$C(AB|AA) = -\log_2 \frac{1+0.01}{2+3^2\times 0.01} = 1.049 \quad (12)$$

which means $C(x||x) = 1.269$ or, in other words, using a xaFCM to represent the sequence x it is possible to compress it relatively to itself using just 1.269 bits.

Calculating the NRC for both compressors we obtain:

- **Using FCM –** $\mathrm{NRC}(x||x) = \frac{2.1272}{6\times\log_2 3} = 0.224$;
- **Using xaFCM –** $\mathrm{NRC}(x||x) = \frac{1.049}{6\times\log_2 3} = 0.110$.

Based on this example, we can infer that, at least for some cases, it is possible to obtain better compression ratios, using xaFCMs instead of traditional FCMs to represent a sequence.

2.2. Parameter selection

2.2.1. Selection of α

Since adjusting the α parameter might not be trivial, as it depends on the choice of d as well as on the alphabet size. It is, however, possible to choose α based on a certain desired probability p for a specific outcome.

In our experiments, in order to avoid having one more parameter to "tweak", we are defining α automatically, in a way such that, if sequence $w \in \mathcal{A}^d$ was only found once after a certain context $c = x_{n-k+1}^{n} \in x$, and no other sequence $\in \mathcal{A}^d$ was found after that context c (in other words, the total of that line, in the model, is 1), we want to be 90% sure that the same situation happens when compressing a sequence relatively to the learned model. In other words, when we calculate the number of bits,

$$-\log_2 P(X_i = t_i|c) \quad (13)$$

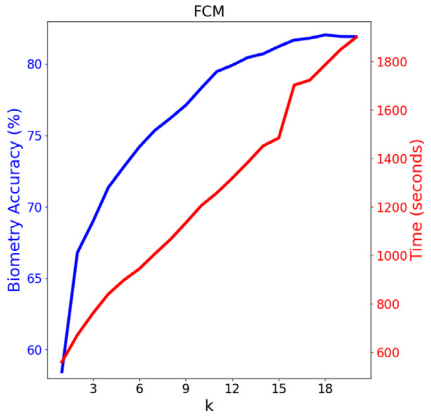

Fig. 1. FCM biometry process: context k changing; the blue line represents the biometry accuracy (in %) and the red line represents the time of execution (seconds). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

needed to compress sequence $t_i = x_{id}, x_{id+1} \ldots x_{id+d-1}$, we want to choose an α such that

$$P(X_i = t_i|c) = 0.9^d. \quad (14)$$

But, since

$$P(w|c) = \frac{\nu(w|c) + \alpha}{\nu(c) + \alpha|\mathcal{A}|^d}, \quad (15)$$

where, we have chosen c and w such that $\nu(c) = \nu(w|c) = 1$. Therefore

$$P(w|c) = 0.9^d \iff \frac{1+\alpha}{1+\alpha|\mathcal{A}|^d} = 0.9^d. \quad (16a)$$

Since $\mathcal{A}^d$ is an extension of $\mathcal{A}$ to d dimensions,

$$\frac{1+\alpha}{1+\alpha|\mathcal{A}|^d} = 0.9^d. \quad (16b)$$

Also, since both the alphabet size $\mathcal{A}$ and the depth d are static parameters, it is easy to solve the equation and choose α in this way. It is also worth mentioning that there is always a possible solution for the equation, since the denominator of the fraction on the left is never equal to zero.

2.2.2. Selection of d

The parameter d is an integer greater or equal to one. As mentioned in Section 2.1, when $d = 1$, we are using a xaFCM which is equivalent to a FCM of the same order k. Therefore, they both produce exactly the same number of bits.

As d increases, so does the RAM needed to store the xaFCM model – but there is not much of an impact (for $d = 11$ the increase in memory usage is about 10%). The reason for the model complexity to only increase this is that the number of different "leaves" in the hash-tables does not change with the choice of d – only the size of each string stored does.

Something to take into account when choosing d is that, the greater the value of d, the harder it would be for an arithmetic encoder to complete its process. Since we only want to compute the NRC, we do not use an encoder. However, to avoid unrealistic results, we want to choose a d that produces an alphabet size of, at most, the MaximumValue(integer)–1 (e.g. $2^{31} - 1$) symbols. For that reason, using an alphabet of size 6, we can say that $1 \leq d \leq 11$.

Often, we are mostly interested in the time it takes to compress a new target sequence, given an already built model representing the reference sequence. With this application in mind, we can say for sure that the d should be as big as possible, since, as mentioned before, less computations need to be done to compress a

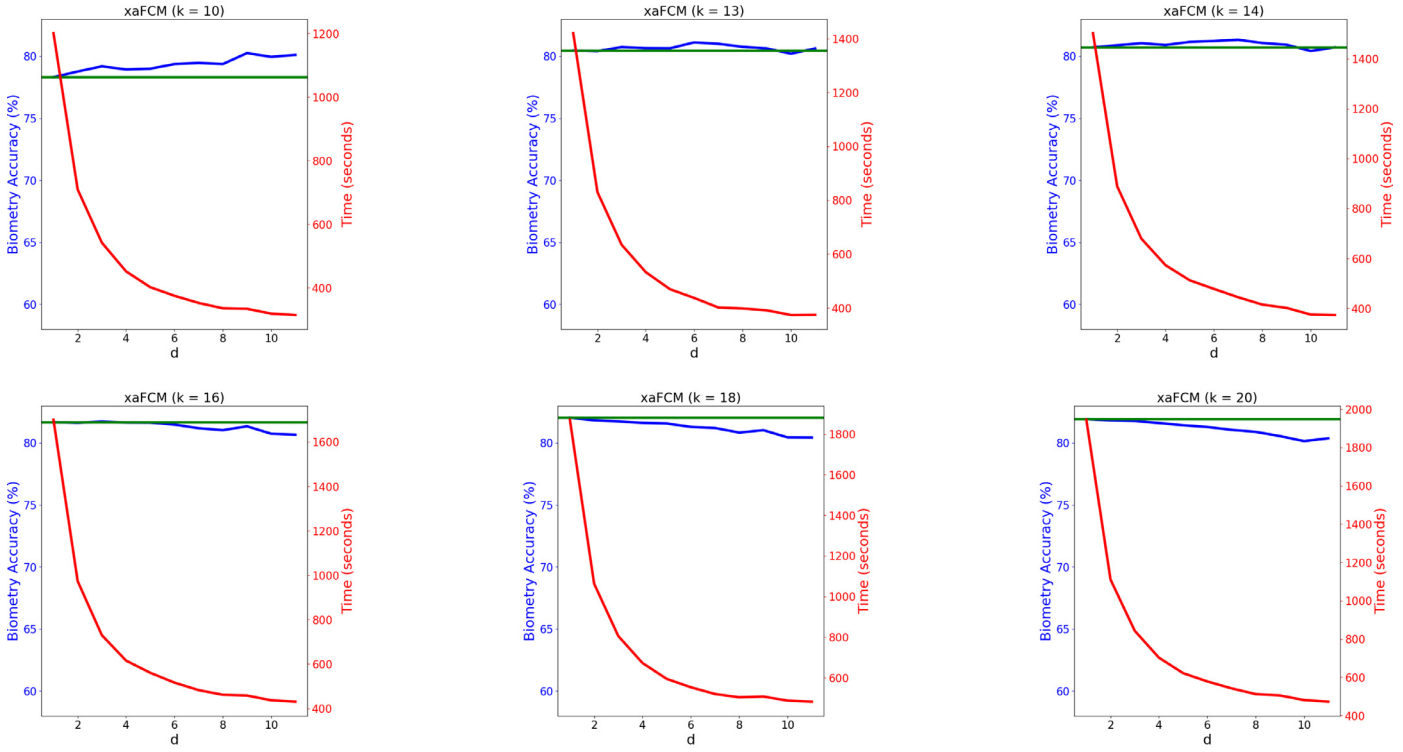

Fig. 2. xaFCM biometry process using high contexts k (fixed); depth d is changing from 1 to 11; the blue line represents the biometry accuracy (in %); the red line represents the time (seconds); the green line represents the accuracy that a FCM with the same context would obtain. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

new target sequence and, therefore, much less time is needed. Results from real experiments can be seen in the next section.

3. Application 1 – ECG biometric identification

In previous works, we have addressed the topic of ECG based biometric identification using a measure of similarity related to the Kolmogorov complexity, called the normalized relative compression (NRC). To attain the goal, we built finite-context models (FCM) to represent each individual [9,10] – a compression-based approach that has been shown successful for different pattern recognition applications [2,4,11]. Other recent works, also based on a compression approach, use the Ziv–Merhav cross parsing algorithm for attaining the same goal [12,13].

Compression-based approaches found in the literature for ECG biometric identification does not seem to take advantage of the fact that the ECG is a quasi-periodical time-series. Since our method uses a semi-fiducial approach (it only detects the R-peak), it is trivial to know where the repetition should happen and take advantage of that fact. From previous results [14], we concluded that, when consecutive *heartbeats*[2] present low levels of noise, their quantization is almost identical. As a consequence of this, we consider that any sequence we analyze is a circular sequence [15]. From this result, it is possible to infer that, compressing the beginning of an heartbeat using the end of the same heartbeat, may be identical to compress it using the end of the previous heartbeat. This may not sound as an advantage, however, this fact allows us to use heartbeats that are not consecutive, when performing the identification of a participant.

[2] For readability, by "heartbeat" we mean the interval between two consecutive R-peaks.

Since the purpose of this paper is to introduce the method, and not to focus too much in the ECG signal, we do not explore this fact. However, it is already being taken it into account when building the algorithm (one of the arguments that the algorithm accepts as input is the length of the expected repetition – i.e. for this application, how many symbols has one heartbeat), because it will be important for building a real system, as we expect more noise to be present and, therefore, some segments need to be discarded when performing the compression [14].

3.1. R-peak detection

The development of a robust automatic *R-peak* detector is essential, but it is still a challenging task, due to irregular heart rates, various amplitude levels and *QRS* morphologies, as well as all kinds of noise and artifacts [16].

We decided to use a *semi-fiducial* method for segmenting the ECG signal and, since this was not the major focus of the work, we used a preexisting implementation to detect *R-peaks*, based on the method proposed in [16]. The reason for using a semi-fiducial approach is that fiducial methods have a higher error of detection, while detecting the R-peaks is, nowadays, an almost trivial process [16]. The method used detects the *R-peaks* by calculating the average points between the *Q* and *S* points (from the *QRS-complexes*) – this may not give the real local maxima corresponding to the *R-peaks*, but it produces a very close point.

For more information regarding the process used for detecting the R-peaks check [16]. The process was already validated by its authors using the standard MIT-BIH arrhythmia database, achieving an average sensitivity of 99.94% and a positive predictivity of 99.96%. It uses bandpass filtering and differentiation operations, aiming to enhance the *QRS* complexes and to reduce out-of-band noise. A nonlinear transformation is used to obtain a positive-

Table 4
CPU time and memory usage (RAM) of the experiments with DNA sequences.

Parameters (context k and depth d)	Average time to learn the Model	Average time to compress	Average memory per model	Total time to run the experiment
$k = 12$, $d = 1$	1649.6 s	1580.5 s	5043.2MB	274.4 h
$k = 12$, $d = 8$	2181.2 s	269.5 s	14350.3MB	59.5 h

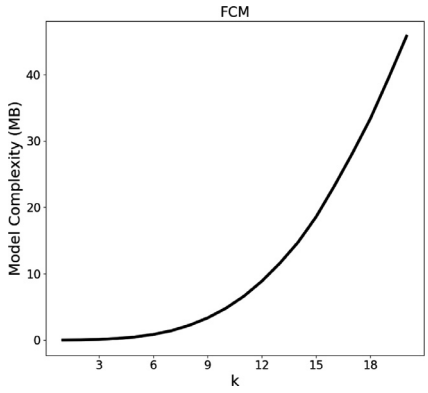

Fig. 3. FCM average model complexity per participant - context k changing.

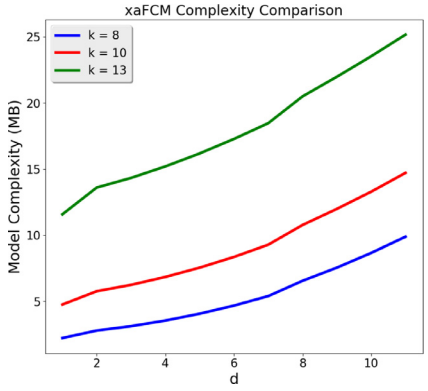

Fig. 4. xaFCM average model complexity per participant - context d is changing; k is fixed.

valued feature signal, which includes large candidate peaks corresponding to the *QRS* complex regions.

3.2. Quantization

Data compression algorithms are symbolic in nature. Text and DNA sequences are well-known examples of symbolic sequences, with well-defined associated alphabets. Contrarily the ECG signal needs first to be transformed into symbols before data compression can be applied.

In this work, we have relied on the SAX (Symbolic Aggregate ApproXimation) representation [17] to transform the ECG into a symbolic time-series.

We consider that the signal is already discrete in the time domain, i.e., that it is already sampled. However, we perform resampling using the previously detected R-peaks.

There is a fundamental trade-off to take into account while performing the choice of the *alphabet size*: the quality produced versus the amount of data necessary to represent the sequence [18]. We tested the experiments using alphabet sizes from 3 up to 20 symbols and using different numbers of symbols each R-R segment (per heartbeat), and found that a combination of using an alphabet size of 6 and 200 symbols per heartbeat produced a good balance between the complexity of the strings/models and the accuracies obtained for biometric identification, allowing us to proceed with the improvement of the compressors, while these parameters remained static. However, this result does not guarantee that the same will hold true for a different dataset or application, nor does it guarantee that these are the optimal parameters. Future work is needed to perform this choice in a more robust and automatic way.

3.3. Experimental results

The database used in our experiments was collected *in house* [10,19], where 25 participants were exposed to different external stimuli – *disgust, fear* and *neutral*. Data were collected on three different days (once per week), at the University of Aveiro, using a different stimulus per day.

The data signals were collected during 25 min on each day, giving a total of around 75 minof ECG signal per participant. Before being exposed to the stimuli, during the first 4 min of each data acquisition, the participants watched a movie with a beach sunset and an acoustic guitar soundtrack, and were instructed to try to relax as much as possible.

By using a database where the participants were exposed to different stimuli, we can check if the emotional state of participants affects the biometric identification process. The database is publicly available for download in.[3]

After all the already explained preprocessing steps are complete, the process in which we perform the biometric identification is the following:

1. Use the complete ECG signals from two days, in order to build a xaFCM model that describes each of the participants;
2. For the remaining day, split the signal, such that each segment has 10 consecutive heartbeats inside it;
3. “Compress” (compute the NRC) each of the segments obtained in the previous step using each of the models obtained in the first step;
4. The model which produces a lowest result is chosen as the candidate for biometric identification.

The justification for the first step is that we do not want to use any information from the ECG of the day where we are trying to perform the ECG biometric identification, since, if we used that information, our results would not match a real situation.

The number of heartbeats needed for ECG biometric identification is undoubtedly useful when building a biometric identification system – any system should ask participants to provide data for identification, using the smallest time interval that is possible, for practical reasons. Based on the results from a previous study [10], we concluded that 10 heartbeats is a good trade-off between collection time (which should be as low as possible) and statistical relevance of the data.

All the experiments were implemented and ran using Python 3.5 (Linux 64 bits) on an Intel(R) Core(TM) i7-6700 CPU @ 3.40 GHz, with 32GB of RAM. For simplicity of code, we have not parallelized the process yet – therefore, only one logical core was used for each experiment.

[3] http://sweet.ua.pt/ap/data/signals/Biometric_Emotion_Recognition.zip.

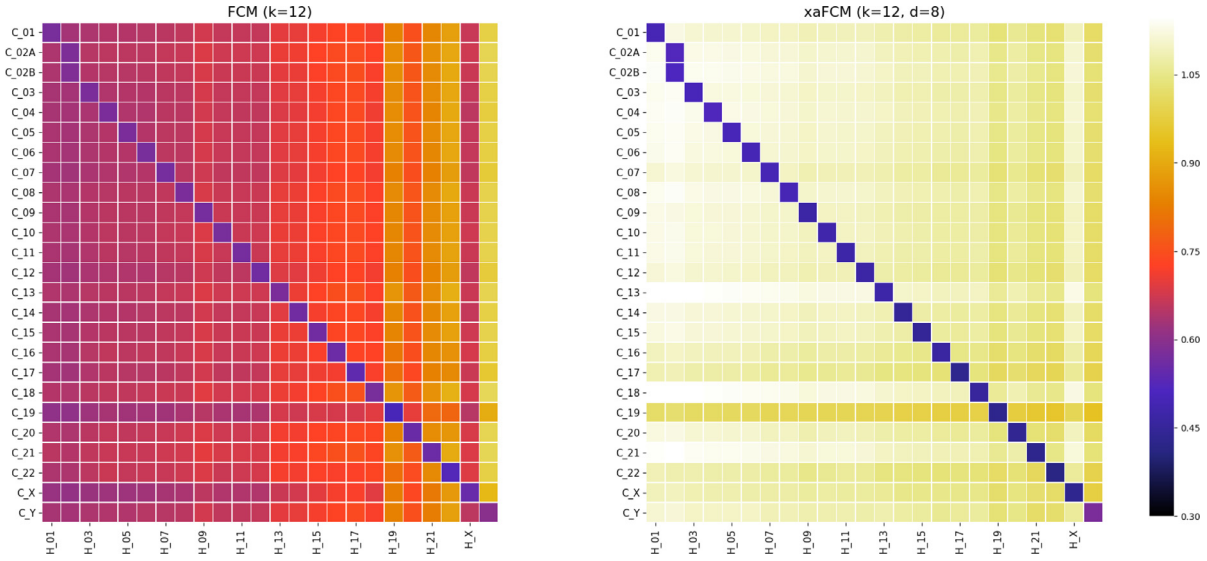

Fig. 5. Normalized compression of the chimpanzee (C) chromosomes relatively to the human (H), using: **(left)** FCM ($k = 12$); **(right)** xaFCM ($k = 12$, $d = 8$).

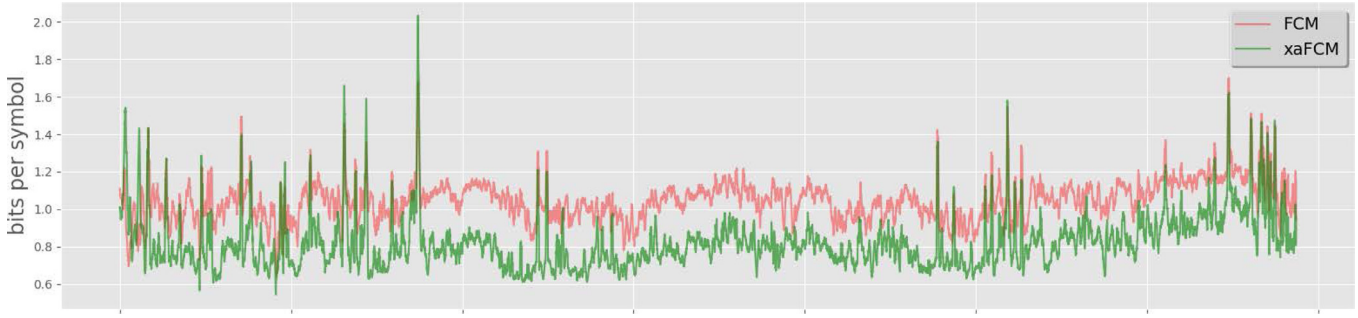

Fig. 6. Profiles of information content of the chimpanzee chromosome 22 relatively to the human chromosome 22 using different models (FCM and xaFCM).

In Fig. 1, it is possible to see a plot with the accuracy obtained for the process described, by using FCM models, with all possible values of k from 1 up to 20. In the red line, it is also possible to see how much time does this process take in total. An important fact is that the time taken to perform the biometry is approximately directly proportional to the size of the context, k, used.

Since the purpose of this paper is to show the appropriateness of xaFCM models, in Fig. 2 are shown six examples of the same experiment, but instead of changing the context k, we have chosen a fixed value of k and tested all possible values of d, the depth of the xaFCMs. From these plots, it is possible to see that the time taken to perform the biometry process for the whole database is up to 3–4 times shorter when using high values of d, having, usually, accuracy ratios comparable with the FCMs of the same order k.

On the experiments using "lower" values for the context k (in this case, $k \leq 14$), it is possible to notice a minor improvement in terms of accuracy as the d increases, at least for the first values of d ($d \leq 7$, more or less). This makes us think that increasing the depth d behaves in a similar way to increasing the depth k of the xaFCM, without the additional cost in terms of testing speed (quite the opposite, actually) and the memory needed does not increase so much as it would by increasing k (Fig. 3).

In higher contexts k we get the same advantages in terms of computing time and memory requirements, however, after a certain point, there is just no real benefit from increasing neither the context k, nor the depth d, since we are looking for "too specific" patterns, that may not appear again on the segments being tested – which, making an analogy to machine learning, we would be overfitting to the training data.

Another aspect we wanted to show, regarding the advantages of using xaFCMs, is the model complexity. In order for the biometric identification to be executed fast, in practice, it is needed to have all the participant models previously loaded into memory. This usually does not pose a problem, if there are not many participants, but it may be useful for building a real biometric identification system.

In Fig. 3, we can see that by increasing the context k of FCM models, the complexity of each model increases exponentially. From our interpretation, a way to avoid this exponential increase is to use an xaFCM with an order slightly lower and increase its depth d. In order to show this, we display the complexity of such models in Fig. 4.

4. Application 2 – DNA sequence relative similarity

An approach for computing the similarity of a sequence relatively to other is to calculate the NRC using one of them as reference and the other as the target. In previous works, this has been done using FCM compressors [2,20–22].

In order to show that the xaFCMs are also suitable for this application, we ran some simulations using the human and chimpanzee DNA sequences, removing the unknown symbols (N). The idea was to use each chromosome of the human species as reference and then compress each chromosome of chimpanzee as the target, using exclusively the model from the reference. Since we know from evolution theory that these two species are closely re-

lated [23], it is expected that, when we are compressing homologous pairs of chromosomes, the NRC should be lower than on the other cases.

To perform the experiment, we used the assembled human chromosomes 1–22, X and Y (3.1GB of data in total) and assembled chimpanzee chromosomes 1, 2a, 2b, 3 to 22, X and Y (3.2GB of data in total).[4] We ran two different simulations: the first one, with a FCM of context $k = 12$; the other with a xaFCM with $k = 12$ and $d = 8$. All the experiments ran on a server with 16-cores 2.13 GHz Intel Xeon CPU E7320 and 256GB of RAM, but the implementation used a single core.

Table 4 shows the average times taken by each experiment, as well as the average memory needed to store the xaFCM model to represent the human chromosomes.

It is clear from these results that the xaFCMs are almost d times faster than an FCM of the same order k. Another advantage is that the memory needed for the xaFCMs does not increase exponentially with d.

The NRC results for the two simulations, with $k = 12$, can be seen in Fig. 5.

It is possible to notice that the heatmap corresponding to the FCM shows better compressions on average. However, using the "perfect" relative compressor, we would expect the NRCs to be as low as possible on the diagonal of the matrix,[5] since they represent related chromosomes. The other squares should have higher NRCs, as they have more variation. This is exactly what happens on the xaFCM test (bottom one in Fig. 5).

This becomes even more clear when we are comparing the compression along the same sequence, as can be seen in Fig. 6.

5. Conclusions and future work

We have shown that xaFCMs are good candidates to represent models for ECG biometric identification. When compared with FCMs, with the same memory usage, better accuracy ratios are usually obtained, using up to around 3–4 times less time to compute the NRCs (depending on the choice of d).

The gains in computational speed increase in the DNA sequence given the higher order of data length. Our experiments show that it is possible to use them for DNA sequence pattern recognition, making them a suitable alternative to the traditional FCMs.

These are promising results, and it seems appropriate to infer that the xaFCMs can be suitable to some other applications, specially when the problem of memory usage or testing speed are crucial. For that reason, in the near future, we plan to test them in different applications, where FCMs have proven suitable, like image pattern recognition [1,11,24] and authorship attribution [4].

Acknowledgments

This work was partially supported by national funds through the FCT - Foundation for Science and Technology, and by European funds through FEDER, under the COMPETE 2020 and Portugal 2020 programs, in the context of the projects UID/CEC/00127/2013 and PTDC/EEI-SII/6608/2014. S. Brás acknowledges the Postdoc Grant from FCT, ref. SFRH/BPD/92342/2013.

[4] All the assembled genome data were downloaded from ftp://ftp.ncbi.nlm.nih.gov/genomes/.

[5] Not exactly the diagonal, because of the second chromosome of the chimpanzee is split into 2a and 2b, making the matrix not a square one.

References

[1] A.J. Pinho, P. Ferreira, Finding unknown repeated patterns in images, in: 19th Eur. Signal Process. Conf. (EUSIPCO 2011), 2011.
[2] D. Pratas, A.J. Pinho, A conditional compression distance that unveils insights of the genomic evolution, in: 2014 Data Compression Conference, IEEE, 2014, pp. 421–422, doi:10.1109/DCC.2014.58.
[3] D.P. Coutinho, M.A.T. Figueiredo, Text classification using compression-based dissimilarity measures, Int. J. Pattern Recognit. Artif. Intell. 29 (05) (2015), doi:10.1142/S0218001415530043.
[4] A.J. Pinho, D. Pratas, P.J.S.G. Ferreira, Authorship attribution using compression distances, in: Data Compression Conference, 2016, doi:10.1109/DCC.2016.53.
[5] D. Pratas, A.J. Pinho, R.M. Silva, J.a. Rodrigues, M. Hosseini, T. Caetano, P. Ferreira, FALCON: a method to infer metagenomic composition of ancient DNA, BioRxiv (2018). http://biorxiv.org/content/early/2018/02/18/267179.abstract.
[6] M. Li, X. Chen, X. Li, The similarity metric, IEEE Trans. Inf. Theory 50 (12) (2004) 3250–3264.
[7] M. Li, P. Vitányi, An Introduction to Kolmogorov Complexity and its Applications, 3rd Ed., Springer, 1997, doi:10.1016/S0898-1221(97)90213-3.
[8] C.H. Bennett, P. Gács, M. Li, Information distance, IEEE Trans. Inf. Theory 44 (4) (1998) 1407–1423. http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=681318.
[9] S. Brás, A.J. Pinho, ECG biometric identification: a compression based approach, in: 2015 37th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC), 2015, pp. 5838–5841, doi:10.1109/EMBC.2015.7319719. http://www.ncbi.nlm.nih.gov/pubmed/26737619.
[10] J.a.M. Carvalho, S. Brás, J. Ferreira, S.C. Soares, A.J. Pinho, Impact of the Acquisition Time on ECG Compression-based Biometric Identification Systems, in: Proceedings of Pattern Recognition and Time Analysis - 8th Iberian Conference (IbPRIA), 2017, pp. 169–176.
[11] A.J. Pinho, P. Ferreira, Image similarity using the normalized compression distance based on finite context models, 18th IEEE Int. Confer. Image Process., 2011.
[12] D.P. Coutinho, H. Silva, H. Gamboa, A. Fred, M. Figueiredo, Novel fiducial and non-fiducial approaches to electrocardiogram-based biometric systems, IET Biom. 2 (2) (2013).
[13] D.P. Coutinho, A. Fred, M. Figueiredo, One-lead ECG-based personal identification using Ziv–Merhav cross parsing, in: Pattern Recognit. (ICPR), 20th Int. Conf., 2010, pp. 3858–3861.
[14] J.M. Carvalho, A.J. Pinho, S. Brás, Irregularity detection in ECG signal using a semi-fiducial method, in: Proceedings of the 22nd RecPad, 2016, pp. 75–76.
[15] R. Grossi, C.S. Iliopoulos, R. Mercas, N. Pisanti, S.P. Pissis, A. Retha, F. Vayani, Circular sequence comparison: algorithms and applications., Algorithms Mol. Biol. 11 (2016) 12, doi:10.1186/s13015-016-0076-6.
[16] P. Kathirvel, M. Sabarimalai, S.R.M. Prasanna, K.P. Soman, An efficient R-peak detection based on new nonlinear transformation and first-order Gaussian differentiator, Cardiovasc. Eng. Technol. 2 (4) (2011) 408–425, doi:10.1007/s13239-011-0065-3.
[17] J. Lin, E. Keogh, S. Lonardi, B. Chiu, 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.
[18] G.C. Rafael, R.E. Woods, Sampling and quantization, Digital Image Processing, Prentice Hall PTR, 2007, doi:10.1002/0470870109.ch3.
[19] J. Ferreira, S. Brás, C.F. Silva, S.C. Soares, An automatic classifier of emotions built from entropy of noise, Psychophysiology (2016), doi:10.1111/psyp.12808.
[20] A.J. Pinho, D. Pratas, P.J.S.G. Ferreira, Bacteria DNA sequence compression using a mixture of finite-context models, in: IEEE Workshop on Statistical Signal Processing Proceedings, 2011, pp. 125–128, doi:10.1109/SSP.2011.5967637.
[21] D. Pratas, A.J. Pinho, Exploring deep Markov models in genomic data compression using sequence pre-analysis, in: European Signal Processing Conference, EUSIPCO 2014,, 2014.
[22] D. Pratas, A.J. Pinho, P.J.S.G. Ferreira, Efficient compression of genomic sequences, in: Data Compression Conference, 2016, doi:10.1109/DCC.2016.60.
[23] A. Hobolth, J.Y. Dutheil, J. Hawks, M.H. Schierup, T. Mailund, Incomplete lineage sorting patterns among human, chimpanzee, and orangutan suggest recent orangutan speciation and widespread selection, Genome Res. 21 (3) (2011) 349–356, doi:10.1101/gr.114751.110.
[24] A.J. Pinho, D. Pratas, P. Ferreira, A new compressor for measuring distances among images, Image Anal. Recognit. 1 (2014) 30–37.