

Multivariate Segmentation of Time Series with Differential Evolution

Daniel Graves¹ Witold Pedrycz²

^{1,2}Department of Electrical & Computer Engineering, University of Alberta, Edmonton, Canada
 Email: dgraves@ualberta.ca¹, pedrycz@ee.ualberta.ca²

Abstract—A new method of time series segmentation is developed using differential evolution. Traditional methods of time series segmentation focus on single variable segmentation and as such often determine sections of the time series with constant slope (i.e. linear). The problem of segmenting multivariate time series is significantly more involved since several time series have to be jointly segmented. Thus the concept of boundary becomes ill-defined since each time series may not be exactly synchronized and change identically in time. The problem is rectified by minimizing the mean of the variance of the slopes determined in each segment. Performance of the method is measured in terms of the classification rate and the accuracy of determination of boundaries. Experimental evidence shows the effectiveness of the method when applied to synthetic and real-world data compared with multivariate time series clustering approaches.

Keywords—Multivariate segmentation, differential evolution, time series, fuzzy clustering.

1 Introduction

Many data come in the form of time series that need to be partitioned into contiguous blocks in time. Tools of Computational Intelligence have been applied to solve the time series partitioning problem are commonly referred to as time series segmentation including fuzzy clustering, hierarchical clustering, and genetic algorithms. Time series segmentation of a single variable has been studied extensively in the literature, c.f. [1,2,3,4,5]. However, there is a lack of research in the area of multivariate time series segmentation or the joint segmentation of several time series.

Only a few multivariate segmentation algorithms found in the literature cluster the time series by taking into account temporal information, e.g., [5,6,7,8]. Unfortunately clustering a time series to perform segmentation poses a few problems. An important issue with segmentation of a time series which has to be addressed is that segments must be contiguous in time. Several approaches have been considered to solve this problem. One method uses constrained clustering to find contiguous segments in time, i.e. constrained hierarchical clustering. Another approach uses time information as an auxiliary component exploited directly in the clustering process. A significant problem is to effectively use this auxiliary information in the clustering procedure leading to boundary identification. However, augmenting clustering with time information is not an appropriate vehicle for segmenting a time series since identification of boundaries requires use of time information as auxiliary information in clustering. A common problem produced by augmented clustering algorithms that account for time information is that the resulting partition does not

necessarily consist of contiguous segments. Currently, the literature lacks an effective method of using the auxiliary information in clustering.

The focus of this research is in multivariate time series segmentation by a more direct segmentation method that avoids the problems produced by using clustering as a vehicle for segmentation by employing differential evolutionary optimization of an objective function. As well, evolutionary methods such as differential evolution are well suited for structural optimization. The reason for using differential evolution (DE) to segmentation a time series is that the segmented boundaries to be optimized cannot be explicitly expressed in the underlying objective function. This makes gradient-based optimization methods not feasible in this problem. The optimization calls for methods of structural optimization and various techniques of Evolutionary Computing are of interest. In this category of methods, DE has shown to be effective as far as quick convergence is concerned.

We focus on detecting slope changes in the time series in order to determine segments. The motivation behind approaching the problem of segmentation by detecting monotonicity is that often a time series can be represented quite effectively using only a handful of monotonic models for each segment. An example is with a set of ECG time series which are often characterized by sharp spikes in the middle. A potential application of segmentation that detects changes in slope is in signal compression, c.f. [2].

There are a number of other applications of multivariate time series segmentation. An example is presented in [9] for process monitoring, diagnosis and control of a medium and high-density polyethylene plant. Another application of multivariate time series segmentation is with music structural segmentation where the objective is to automatically determine a partition of structural elements in music, i.e. verse, chorus, bridge, by detecting significant transition points in the music.

The material is organized in five sections. A literature review is covered in Section 2. Section 3 includes a description of the segmentation algorithm and presents the essentials of differential evolution. Section 4 concentrates on the experimental evaluation of the multivariate segmentation algorithm. The proposed method is evaluated on synthetic data sets and on several real-world time series data sets coming from Time Series Data Library [10]. Section 5 presents the conclusions.

2 Literature Review

Time series segmentation is a very important topic with temporal information processing as long time series need to be broken down into relevant segments. The problem of segmentation has not been solved with high boundary detection accuracy in the literature hence most of research conducted in the literature focuses on the segmentation aspect and leaves the labelling of segments to be done later.

Most time series segmentation approaches look at the problem of segmenting a time series of a single variable. There are three main categories of classical segmentation algorithms available in the literature:

- Bottom-up
- Top-down
- Sliding-window

Unfortunately high segmentation accuracy is not always achieved and there is no one method preferred than the others. The three algorithms are described in [1]. All three have an error threshold where they stop segmenting once the error is exceeded. The first bottom-up starts with each sample in its own segment much like agglomerative hierarchical clustering. Likewise top-down is similar to divisive hierarchical clustering in that the entire time series is broken down into smaller segments until the error for each segment is less than the error threshold. Sliding-window is a real-time segmentation algorithm that starts with the first sample and extends the segment with each new sample until an error threshold is exceeded.

Determining segments via linear approximation and evolutionary methods is common in the literature such as methods using genetic algorithms to approximate an ECG time series through a set of its linear segments, cf. [2]. In [2], the fitness function minimizes the variability between the maximum and minimum slopes for each consecutive sample within the proposed segment. The number of segments is determined beforehand. The experiments are conducted on ECG signals which show fairly good match on the data, i.e. little information appears to be lost during segmentation (which also leads to dimensionality reduction).

Multivariate time series are common in practical problems. Many of the approaches discussed in the literature are not directly applicable to them. An inherent problem with multivariate time series segmentation is that it is difficult to determine boundary points where each feature time series in the multivariate time series unanimously agree. For example, a question arises whether a significant change in one feature's value where all other features do not change significantly constitutes a boundary change point for a time series. Also not all features may change at exactly the same point in time; hence, multivariate segmentation approaches require greater boundary point flexibility to account for the latent changes among the feature values. As a result, some limited work has been done in the literature directly applicable to multivariate time series segmentation, c.f. [5,6,7,9].

The multivariate time series segmentation method developed in [9] approaches the problem as a constrained clustering problem with the constraint that a cluster (i.e.

segment) must contain patterns at successive time points. The approach also constructs fuzzy sets for specifying the segments detected with fuzzy boundaries. Local Principal Component Analysis (PCA) models are used in clustering of the time series into segments based on Gath-Geva clustering. They apply the segmentation algorithm to process monitoring, diagnosis and control in particular monitoring a medium and high-density polyethylene plant, i.e. production of a versatile plastic. The data is multivariate and is successfully segmented via the proposed clustering algorithm.

A Gath-Geva-based clustering algorithm similar to the approach in [7,9] is introduced for application to multivariate time series segmentation. The approach described in [9] and [7] utilize a modified Gath-Geva clustering based on the probabilistic PCA (PPCA) [11] for each cluster. The method in [9] and [7] constrain the segment (cluster) fuzzy sets (membership matrix) to Gaussian membership functions. The method in [7] uses modified Gath-Geva clustering (a.k.a. fuzzy maximum likelihood clustering) to cluster a multivariate time series. The time information is included as an attribute in the time series. The segments are projected onto principal components based on a localized PPCA models for the rationale that the correlation between variables often changes with multivariate time series. The distance measure in the clustering is chosen to include two parts: Gaussian fuzzy sets determining distance in the time domain and a second term that measures the distance of the data from the PCA hyperplane model. The number of principal components to keep in the PPCA model and the number of segments (clusters) are two important parameters that need to be selected via some other method. The authors include some case studies in the literature for the application of the approach to multivariate time series segmentation although the experimental evidence is limited.

Speaker segmentation is discussed in [5]. The method exploits the BIC (Bayesian Information Criterion) procedure, known also as the minimum description length (MDL). The objective is to determine segments of different speakers. Some improvements are found in their approach over traditional BIC approaches. Multivariate segmentation in [6] is accomplished via constrained clustering for the purpose to determine the important structural components of a musical recording, i.e. intro, verse, chorus, solo, bridge, outro. They report some improvements in the classification rate and boundary f-value with their method over traditional approaches. A multivariate time series segmentation algorithm based on hierarchical Bayesian clustering was developed in [12] specifically designed for astronomical data that can be modelled as a Poisson counting process. Some experiments on synthetic data and real astronomical data are provided however they are limited. Authors in [8] describe multivariate segmentation approach for motion data using singular value decomposition (SVD) in the distance measure between multivariate motion time series. The experimental results are very limited.

3 Multivariate Segmentation

3.1 Problem Formulation

The problem of multivariate segmentation involves finding the boundaries in a time series that denote changing behaviour in the time series. For instance a time series might be increase steadily and then at a time t_0 the time series changes behaviour and starts decreasing steadily. Extending this to the multivariate case, each variable which is a time series in itself, exhibits changes in behaviour after the boundary at t_0 . With single-variable time series segmentation, segments are contiguous homogenous partition series that are specified by a set of boundaries that divides segments from their adjacent ones. Multivariate segmentation is the problem of finding a set of boundaries that jointly divides each time series into a set of contiguous partitions of multivariate time series. With multivariate time series, segmentation is more complicated since each feature can be considered a separate time series and hence what might be considered a boundary in one may not be considered a boundary in the other. i.e., the transitions in each feature signal may start and end at different points in time providing conflicting clues as to where the segment starts and ends.

Given a multivariate time series $x_{t,k}$ for $t=1 \dots T$ with M features $k=1 \dots M$, the time series can be segmented through slope analysis of each feature. The idea is to find segments with similar slopes, i.e. finding segments that are monotonic with relatively constant slopes. Assuming the number of segments c is known a priori, the boundaries for the segments are the free-parameters given by $[b_1, b_2, \dots, b_{c-1}]$ where $b_0 = 1$ and $b_c = T$. An example of a multivariate segmentation is given in figure 1.

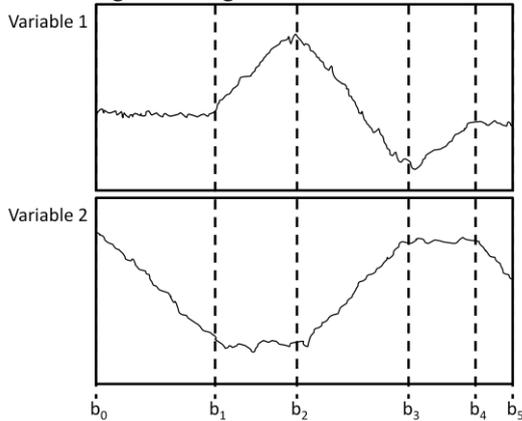


Figure 1: Multivariate segmentation illustration

3.2 Differential Evolution

Differential evolution is simple population-based evolutionary optimization technique that minimizes (or maximizes) an objective function much like genetic algorithms, c.f. [13]. Differential evolution uses mutation and crossover operations just as genetic algorithms; however, the mutation operation is based on population vector differences. The population of vectors are given by the set $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_P\}$ where each vector \mathbf{x}_i is real-valued and has dimensionality M . The objective function to

be minimized is denoted by $f(\mathbf{x})$. The vectors in the population assume values in the unit interval and P is the size of the population. The range of values for the population vectors is on the interval $[0,1]$.

The essence of the differential evolution is expressed via the following pseudo code

1. Initialize Population of size P
2. Loop until stopping criterion met
 1. For each element \mathbf{x}_i in the population
 1. Randomly select three parameter vectors: $\mathbf{x}_{r1}, \mathbf{x}_{r2}, \mathbf{x}_{r3}$ (not the same)
 2. Generate a mutant vector $\mathbf{v}_i = \mathbf{x}_{r1} + G[\mathbf{x}_{r2} - \mathbf{x}_{r3}]$ [Mutation]
 3. Generate trial vector $\mathbf{u}_i = \text{crossover}(\mathbf{v}_i, \mathbf{x}_i)$ by mixing vectors \mathbf{v}_i and \mathbf{x}_i [Crossover]
 4. If $f(\mathbf{u}_i) < f(\mathbf{x}_i)$ then replace \mathbf{x}_i with \mathbf{u}_i in population
3. Termination

The difference factor G controls the intensity of mutation (i.e. vector differences). The difference factor G can take on positive real numbers as well as zero, c.f. [4]. There is also a crossover operation. Many different kinds of crossover operations are possible. Here we use the one presented in [4]. This crossover operation combines two vectors \mathbf{x}_i and \mathbf{v}_i according to the pre-specified probability p_{cr} . The resulting crossover vector produced by crossover is give by the expression

$$u_{i,k} = \begin{cases} v_{i,k} & \text{if } r_k < p_{cr} \\ x_{i,k} & \text{otherwise} \end{cases} \quad (1)$$

where r_k is a uniform random sequence of numbers on the interval $[0,1]$ for $k=1 \dots M$.

3.3 Fitness Function

The boundaries of the segments are determined via differential evolution. Here we elaborate on the pertinent algorithmic details.

A difference (describing slope) sequence is calculated for each variable $k=1 \dots M$ in the multivariate time series

$$D_{tk} = x_{t+1,k} - x_{t,k} \quad (2)$$

for $t=1 \dots T-1$.

Given a set of boundaries $[b_1, b_2, \dots, b_{c-1}]$ where $b_0 = 1$ and $b_c = T$, the length of each segment is given as

$$L_i = b_i - b_{i-1} \quad (3)$$

The variance of the slopes of each variable are calculated via the expression

$$S_{ik}^2 = \sum_{t=b_{i-1}}^{b_i} \frac{(D_{tk} - \bar{D}_{ik})^2}{L_i - 1} \quad (4)$$

where $\bar{D}_{ik} = 1/L_i \sum_{t=b_{i-1}}^{b_i} D_{tk}$ is the mean slope of each variable $k=1 \dots M$ for each segment $i=1 \dots c$. The mean of the standard deviations across the variables is calculated for each segment, i.e.

$$S_i = \frac{1}{M} \sum_{k=1}^M \sqrt{S_{ik}^2} \quad (5)$$

for $i=1 \dots c$.

Finally multivariate segmentation is accomplished by minimizing the following fitness function

$$F = \sum_{i=1}^c S_i L_i \quad (6)$$

where S_i is the standard deviation defined in (4) and L_i is the segment length defined in (2). The resulting segmentation favors segments with slopes that do not vary greatly; hence, the resulting segments are approximately linear.

3.4 Population Encoding

Each population vector is encoded in differential evolution as a vector of real-valued numbers denoted by $\mathbf{y} = [y_1, y_2, \dots, y_{c+1}]$. The set of boundaries $\mathbf{b} = [1, b_1, b_2, \dots, b_{c-1}, T]$ is calculated from the vector \mathbf{y} where the first and last elements ($b_0 = 1$ and $b_c = T$) can be discarded. The boundaries are calculated in the form:

1. $\mathbf{z} = \text{sort} - \text{least} - \text{to} - \text{greatest}(\mathbf{y})$
2. $\mathbf{a} = \frac{\mathbf{z} - z_1}{z_{c+1}}$
3. $\mathbf{b} = \text{round}(1 + (T - 1)\mathbf{a})$

The elements of vector \mathbf{y} is sorted from least to greatest and its sorted version is denoted as a new vector \mathbf{z} so that the boundaries are given in order. The vector \mathbf{z} is normalized to a vector of numbers on the range $[0,1]$ called \mathbf{a} by subtracting the smallest element in \mathbf{z} given by z_1 since the vector is sorted and divided by the largest element z_{c+1} . The vector \mathbf{a} maps directly to boundaries in the time series where 0 is the start of the time series and 1 is the end of the time series. The vector \mathbf{b} is calculated by expanding the range of \mathbf{a} to the set of integer time coordinates $\{1, T\}$. The reason for sorting the vector is that taking vector differences (the mutation operation in differential evolution) can result in negative values. Simply normalizing the vector by its sum as done in [2] would produce erroneous results since vector elements are not necessarily positive. Elements of \mathbf{y} needed to be mapped to the interval $[0,1]$ where the smallest value of \mathbf{y} is mapped to the start of the time series.

4 Experiments

The objective function described in section 3.2 minimizes slope variance for each segment. The optimization procedure is unsupervised, i.e. does not use pre-labelled data during optimization. The following synthetic experiments are evaluated against an *a priori* segmentation provided by a human expert. The classification rate and boundary f-value criteria are used to evaluate the synthetic experiments. The classification rate is the percentage of correctly classified samples according to a pre-segmented time series (human labelled) and is indicative of the precision. The boundary f-value is a measure of the number of correct boundaries in the resulting segmentation. The boundaries discovered by the algorithm are denoted by vector \mathbf{b} and the human labelled boundaries are given by the vector \mathbf{h} where \mathbf{b} and \mathbf{h} have length denoted by L_b and L_h respectively. The boundary f-value is determined according to the following expression

$$f = \frac{2pr}{p+r} \quad (7)$$

where p is the boundary precision and r is the boundary recall calculated as follows

$$n = \sum_{i=1}^{L_h} \begin{cases} 1 & \text{if } \theta > \left| \mathbf{h} - \min_{\forall j=1 \dots L_b} \{ |h_i - b_j| \} \right| \\ 0 & \text{otherwise} \end{cases} \quad (8)$$

$$p = \frac{n}{L_b} \quad (9)$$

$$r = \frac{n}{L_h} \quad (10)$$

where n is the number of correctly classified boundaries with respect to the provided human labelled boundaries and θ is a threshold value which determines the tolerance or region of acceptable difference between the human labelled boundary and machine labelled boundary. The following figure demonstrates the calculation of the f-value.

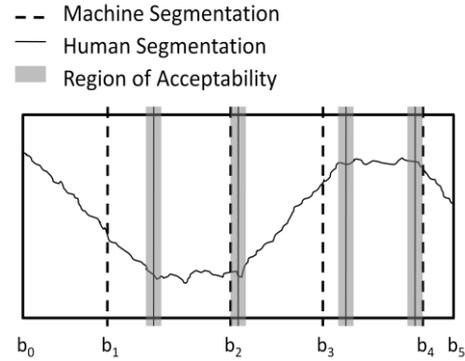


Figure 2: Multivariate segmentation illustration

The reconstruction error is the performance index for the real-world data sets since no human labels are present, i.e.

$$E = \frac{1}{MT} \sum_{k=1}^M \sum_{t=1}^T (\hat{x}_{t,k} - x_{t,k})^2 \quad (11)$$

where $\hat{x}_{t,k}$ is the reconstructed time series. The time series $\hat{x}_{t,k}$ is reconstructed by approximating each segment via a linear line of the form $\hat{x}_{t,k} = m_k t + b_k$ from the first point in the segment to the last point in the segment for each variable $k=1 \dots M$. The values m_k and b_k are the slope and intercept, respectively, of the line for each variable k .

Four synthetic time series were segmented using the described multivariate time series segmentation algorithm. Each variable for time series DATA1, DATA2, DATA3, and DATA4 are given in the figure below where DATA1 and DATA3 have four feature time series and DATA2 and DATA4 have three. The length of each time series is 325, 300, 775, and 700 samples respectively. These time series were generated by creating a collection of linear segments and then adding white noise.

The experimental results on the synthetic data are shown in table 1. They are compared with FCM-DFS (Fuzzy C-Means clustering with Distinct Feature Sets) time series clustering described in [14] for segmentation. The value of theta was chosen to be relatively small, i.e. 3, since the synthetic data sets are relatively simple often with an underlying linear component mixed with white noise. Also, it was noted that generally the segmented boundaries were either very close to the human defined boundary (within 3 samples) or the boundary was not properly discovered by the segmentation algorithm.

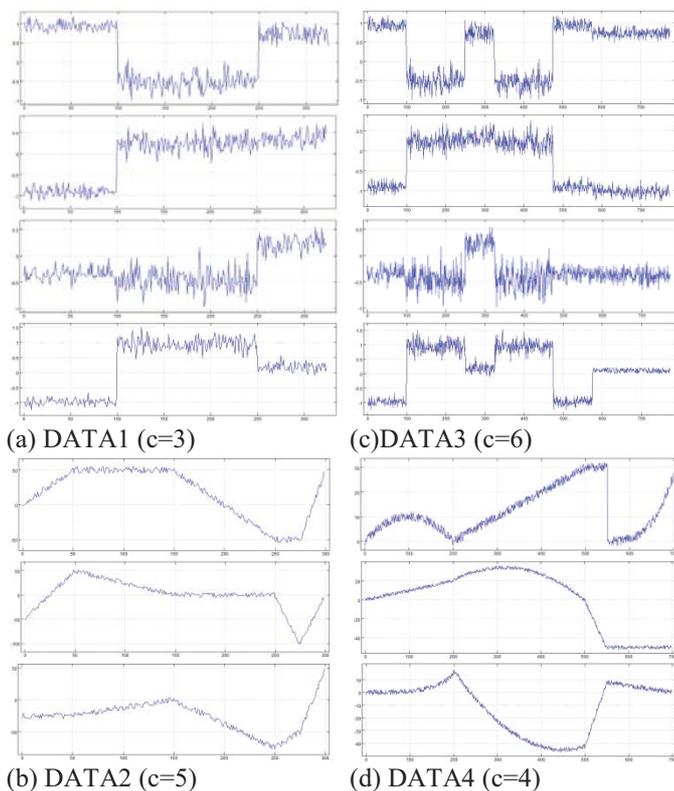


Figure 3: Synthetic Data

The following figure shows the convergence of differential evolution.

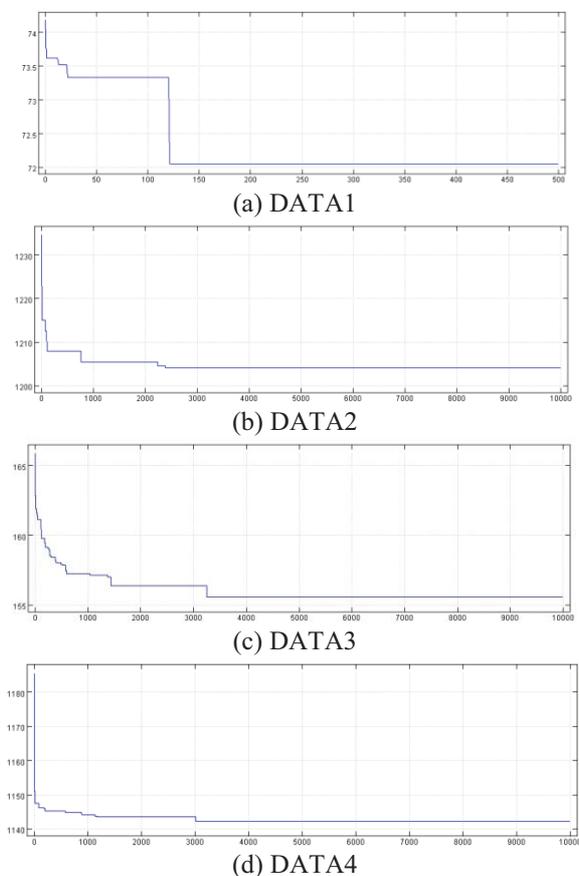


Figure 4: Fitness function in successive generations of differential evolution

Table 1: Synthetic experimental results

	P	G	p_{cr}	Num Iter.	F-value ($\theta = 3$)	Classification Rate
DATA1	100	1	0.25	500	100%	99.7%
DATA2	100	0.1	0.35	10000	50%	67.3%
DATA3	100	0.1	0.3	10000	100%	99.6%
DATA4	100	0.1	0.35	10000	66.7%	57.3%

The differential evolution parameters were fairly robust meaning that changes in the crossover rate and differential gain did not change performance dramatically. However, as the time series became more complex, i.e. DATA3, the differential evolution parameters had to be tuned to prevent the method from being trapped in local minima. From the practical point of view, the essential differential evolution parameters are differential gain, and crossover probably. It was found that increasing population size beyond 100 did not greatly improve the final boundaries. Initially the crossover probably was set to 0.5 and differential gain was set to 1. It was found through parameter adjustments that performance improved with slightly smaller crossover probability. A differential gain of 1 was sufficient on simple data sets; however, more complicated data sets such as DATA2, DATA3, and DATA4 required a smaller value for differential gain (i.e. less mutation) in order to obtain better classification rates and boundary f-value.

Table 2: Synthetic experimental results with FCM-DFS

	m	α	F-Value ($\theta = 3$)	Classification Rate
DATA1	2	0.5	66.7%	99.7%
DATA2	2	0.1	25.0%	80.3%
DATA3	2	0.4	71.4%	99.7%
DATA4	1.3	0.5	16.1%	81.2%

The accuracy of the boundaries discovered by FCM-DFS is not as high when compared with differential evolutionary multivariate segmentation on these synthetic data sets. The classification rate is higher for FCM-DFS but since the boundary f-value is much higher for the proposed method, the segmentation with DE is much more accurate with the determined boundary locations. The reason for the smaller classification rate with DE is that in some of the data sets, only one boundary is determined correctly and its location is way-off the expected value. With FCM-DFS, several boundaries are not determined very well (accuracy is very low) but the locations are a bit closer to their expected values when a boundary is missed. The experiments on FCM-DFS were run for several different values of m {1.3, 1.5, 1.7, 2, 2.3, 2.5, 3} and several values of α {0.05, 0.1, 0.2, 0.4, 0.5, 0.6, 0.8, 0.9, 0.95}.

The segmentation approach was applied to several real valued time series [10]. The performance of the segmentation is evaluated against reconstruction error (11) as shown in Table 3.

The number of clusters was based on a visual estimate of the number of segments from the plot and then running several different values of c in close proximity to the estimated number of clusters. The results show that the segmentation algorithm does fairly well on some of the data sets with small reconstruction errors especially on power station and hog data sets.

Table 3: Real-world time series experimental results

	P	G	p_{cr}	Num Iter.	c	Reconstruction Error
Lynx Pelts	100	0.1	0.9	4000	13	0.1253
Power Station	100	0.5	0.3	4000	12	0.0917
Hog	100	0.1	0.9	4000	16	0.0940
Housing	100	0.5	0.3	4000	17	0.1386
Housing Starts	100	2	0.3	4000	23	0.2185

5 Conclusions

The results on synthetic data show that multivariate time series segmentation show promise as a multivariate time series segmentation tool. The results are quite good on noisy synthetic data with boundary f-values reaching up to 100% and classification rates around 99%. Parameter tuning was important especially as the time series become more complicated and required smaller differential gain values, smaller crossover probability values and more iterations to achieve best results.

Future directions in research involve applying the multivariate segmentation algorithm to the problem of musical segmentation – a particularly difficult problem that automatically finds a partition in music that corresponds to the structural labels of a song, i.e. intro, verse, chorus, bridge, outro. Replacing the linear segment models with more flexible autoregressive models will be developed in future segmentation algorithms.

Acknowledgment

Support from the NSERC and Canada Research Chair (CRC) Program (W. Pedrycz) is gratefully acknowledged. This research is supported by a post-graduate PhD iCore fellowship (D. Graves).

References

- [1] E. Keogh, S. Kasetty, On the need for time series data mining benchmark: a survey and empirical demonstration, *Proc. of the 8th ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining*, Edmonton, Alberta 102-111, 2002.
- [2] A. Gacek, W. Pedrycz, A genetic segmentation of ECG signals, *IEEE Trans. On Biomedical Engineering*, 50(10): 1203-1208, 2003.
- [3] F.L. Chung, T.C. Fu, V. Ng, R.W.P. Luk, An evolutionary approach to pattern-based time series segmentation, *IEEE Trans on Evolutionary Computation*, 8(5): 471-489, 2004.
- [4] J.A. Crossman, H. Guo, Y.L. Murphey, J. Cardillo, Automotive signal fault diagnostics – part I: signal fault analysis, signal segmentation, feature extraction and quasi-optimal feature selection, *IEEE Trans on Vehicular Technology*, 52(4): 1063-1075, 2003.
- [5] P. Delacourt, C.J. Wellekens, DISTBIC: a speaker-based segmentation for audio data indexing, *Speech communications*, 32: 111-126, 2000
- [6] M. Levy, M. Sandler, Structural segmentation of music audio by constrained clustering, *IEEE Trans. on Audio, Speech, and Language Processing*, 16(2): 318-326, 2008.
- [7] J. Abonyi, B. Feil, S. Nemeth, P. Arva, Modified Gath-Geva clustering for fuzzy segmentation of multivariate time-series, *Fuzzy Sets and Systems*, 149(1): 39-56, 2005.
- [8] C. Li, Z. Peng, S.Q. Zheng, B. Prabhakaran, Segmentation and recognition of multi-attribute motion sequences, *Proc ACM Int. Conf. Multimedia*, New York, 836-843, Oct 2004.

- [9] J. Abonyi, B. Feil, S. Nemeth, P. Arva, Fuzzy clustering based segmentation of time-series, *Proc. Intelligent Data Analysis*, 2810: 275-285, 2003.
- [10] R.J. Hyndman, *Time Series Data Library*, <http://www.robjhyndman.info/TSDL> (accessed on Jan 29, 2009).
- [11] M.E. Tipping, M. Bishop, Mixtures of probabilistic principal component analyzers, *Neural Computation*, 11(2): 443-482, 1999.
- [12] N. Dobigeon, J.Y. Tourneret, J.D. Scargle, Joint segmentation of multivariate astronomical time series Bayesian sampling with a hierarchical model, *IEEE Transactions on Signal Processing*, 55(2): 414-423, 2007.
- [13] R. Storn, K. Price, Differential evolution – a simple and efficient heuristic for global optimization over continuous spaces, *Journal of Global Optimization*, 11: 341-359, 1997.
- [14] D. Graves, W. Pedrycz, Structural segmentation of music with fuzzy clustering, *In Proc. Acoustics Week in Canada*, Vancouver, BC, Canadian Acoustics, Oct 2008.